



**TUGAS AKHIR – TE 141599**

**PERANCANGAN JARINGAN SARAF TIRUAN UNTUK  
PENGENALAN POLA PADA NAVIGASI *MOBILE ROBOT*  
BERBASIS DATA GAMBAR**

**Pauzi Saputra  
NRP. 2215 105 028**

**Dosen Pembimbing  
Ir. Rusdhianto Effendi AK., MT.  
Ir. Joko Susila, MT.**

**DEPARTEMEN TEKNIK ELEKTRO  
Fakultas Teknologi Elektro  
Institut Teknologi Sepuluh Nopember  
Surabaya 2017**

*Halaman ini sengaja dikosongkan*



***FINAL PROJECT - TE 141599***

***ARTIFICIAL NEURAL NETWORK DESIGN FOR PATTERN  
RECOGNITION OF MOBILE ROBOT NAVIGATION BASED ON  
IMAGE DATA***

**Pauzi Saputra  
NRP. 2215 105 028**

***Supervisor***  
**Ir. Rusdhianto Effendi AK., MT.**  
**Ir. Joko Susila, MT.**

***DEPARTEMENT OF ELECTRICAL ENGINEERING***  
***Faculty Of Electrical Technology***  
**Institut Teknologi Sepuluh Nopember**  
**Surabaya 2017**

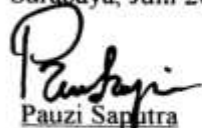
*Halaman ini sengaja dikosongkan*

## PERNYATAAN KEASLIAN TUGAS AKHIR

Dengan ini saya menyatakan bahwa isi sebagian maupun keseluruhan Tugas Akhir saya dengan judul **“Perancangan Jaringan Saraf Tiruan Untuk Pengenalan Pola Pada Navigasi *Mobile Robot* Berbasis Data Gambar”** adalah benar-benar hasil karya intelektual mandiri, diselesaikan tanpa menggunakan bahan-bahan yang tidak diijinkan dan bukan merupakan karya pihak lain yang saya akui sebagai karya sendiri.

Semua referensi yang dikutip maupun dirujuk telah ditulis secara lengkap pada daftar pustaka. Apabila ternyata pernyataan ini tidak benar, saya bersedia menerima sanksi sesuai peraturan yang berlaku.

Surabaya, Juni 2017



Pauzi Saputra

NRP. 2215 105 028

*Halaman ini sengaja dikosongkan*

**PERANCANGAN JARINGAN SARAF TIRUAN UNTUK  
PENGENALAN POLA PADA NAVIGASI MOBILE ROBOT  
BERBASIS DATA GAMBAR**

**TUGAS AKHIR**

**Diajukan Guna Memenuhi Sebagian Persyaratan  
Untuk Memperoleh Gelar Sarjana Teknik  
Pada**

**Bidang Studi Teknik Sistem Pengaturan  
Departemen Teknik Elektro  
Institut Teknologi Sepuluh Nopember**

**Menyetujui:**

**Dosen Pembimbing 1,**

**Dosen Pembimbing 2,**

  
**Ir. Rusdhianto Effendi AK., MT.**  
**NIP: 195704241985021001**

  
**Ir. Joko Susila, MT.**  
**NIP: 196606061991021001**



*Halaman ini sengaja dikosongkan*



# **PERANCANGAN JARINGAN SARAF TIRUAN UNTUK PENGENALAN POLA PADA NAVIGASI *MOBILE ROBOT* BERBASIS DATA GAMBAR**

**Nama** : Pauzi Saputra  
**Pembimbing I** : Ir. Rusdhianto Effendi AK., MT.  
**Pembimbing II** : Ir. Joko Susila, MT.

## **ABSTRAK**

Perkembangan ilmu teknologi khususnya pada sistem serba otomatisasi yang salah satunya melibatkan peran robotika pada kendaraan semakin terbuka lebar. Mobile robot merupakan salah satu bentuk prototype. Penerapan sistem cerdas pada robot dibutuhkan karena robot diberi pilihan sesuai navigasi. Navigasi sebagai petunjuk dari arah robot beroperasi. Untuk menerapkan sistem navigasi tersebut maka diterapkan pengenalan pola dari obyek yang dijadikan navigasi robot. Tujuan pengenalan pola adalah untuk mengenali karakteristik benda. Pengelatan pola benda dilakukan berkali-kali terhadap robot untuk melatih kecerdasan robot dalam melakukan pengidentifikasian obyek navigasi. Kemudian robot diberi tugas untuk mengelompokkan dari bentuk yang telah dikenali sebagai pembeda arah tiap navigasi. Navigasi berupa obyek yang direkam melalui kamera robot sebagai indra penglihatan buatan. Obyek bisa dibedakan denan obyek lain dengan menentukan batas warna biru, dan selain biru tidak dijadikan bentuk navigasi. Pada penelitian tugas akhir sistem diimplementasikan pada mobile robot Qbot dan hasil tersebut disimulasikan dalam bentuk file trajectory luasan operasi robot menjadi titik koordinat robot beroperasi sebagai pembanding maupun interface saat robot dioperasikan. Pergerakan mobile robot dapat lebih leluasa monuver sesuai jalur itu disebabkan sistem terhubung jaringan wireless dengan komputer sebagai host-nya.

Kata kunci : Pengenalan Pola, Sistem Cerdas, *Mobile robot*, Jalur, Navigasi, dan Posisi.

*Halaman ini sengaja dikosongkan*

# **ARTIFICIAL NEURAL NETWORK DESIGN FOR PATTERN RECOGNITION OF MOBILE ROBOT NAVIGATION BASED ON IMAGE DATA**

**Name** : Pauzi Saputra  
**Supervisor** : Ir. Rusdhianto Effendi AK., MT.  
**Co-Supervisor** : Ir. Joko Susila, MT.

## **ABSTRACT**

*Technological developments, especially in automated systems that one of them involves the role of robotics in the vehicle is getting wide open. Mobile robot is one form of prototype. Implementation of intelligent systems on the robot is needed because the robot is given the choice according to navigation. Navigation as a hint of the direction the robot operates. To apply the navigation system is then applied pattern recognition of the objects used as navigation robot. The purpose of pattern recognition is to recognize the characteristics of objects. The patterning of objects is done repeatedly to the robot to train the robot intelligence in identifying the navigation object. Then the robot is given the task to classify from a form that has been recognized as a differentiator of each navigation. Navigation in the form of objects recorded through the robot camera as a sense of artificial sight. Objects can be distinguished by other objects by specifying the blue boundary, and in addition to the blue is not used as a form of navigation. In the final project, the system is implemented in mobile robot Qbot and the result is simulated in the form of trajectory file of the robotic operation area into the coordinate point of the robot operating as a comparison and the interface when the robot is operated. Movement of the mobile robot can more freely maneuver according to the path is due to the system connected wireless network with the computer as its host.*

**Keywords:** *Pattern Introduction, Intelligent System, Mobile robot, Track, Navigation, and Position.*

*Halaman ini sengaja dikosongkan*

## KATA PENGANTAR

Puji syukur kepada Tuhan Yang Maha Esa atas rahmat dan karunia-Nya penulis dapat menyelesaikan Tugas Akhir ini dengan baik yang berjudul :

**“Perancangan Jaringan Saraf Tiruan Untuk Pengenalan Pola  
Pada Navigasi *Mobile Robot* Berbasis Data Gambar”**

Tugas Akhir ini merupakan persyaratan dalam menyelesaikan pendidikan program Strata-Satu di Jurusan Teknik Sistem Pengaturan, Departemen Teknik Elektro, Fakultas Teknologi Elektro, Institut Teknologi Sepuluh Nopember (ITS) Surabaya.

Tugas Akhir ini dibuat berdasarkan teori-teori yang didapat selama mengikuti perkuliahan, berbagai literatur penunjang dan pengarahan dosen pembimbing dari awal hingga akhir pengerjaan Tugas Akhir ini. Pada kesempatan ini, penulis ingin berterima kasih kepada pihak-pihak yang membantu dalam menyelesaikan tugas akhir ini, khususnya kepada:

1. Ibu, Bapak, kakak dan seluruh keluarga yang memberikan dukungan dengan baik.
2. Ir. Rusdhianto Effendi AK., MT. selaku dosen pembimbing 1 atas bimbingan dan arahan selama penulis mengerjakan tugas akhir ini dan juga telah meminjamkan ruangan laboratorium sistem dan sibernetika.
3. Ir. Joko Susila, MT. selaku dosen pembimbing 2 atas bimbingan dan arahan selama penulis mengerjakan tugas akhir ini.
4. Seluruh dosen bidang studi Teknik Sistem Pengaturan.
5. Teman-teman teknik sistem pengaturan yang menyelesaikan tugas akhir di laboratorium sistem dan sibernetika yang tidak dapat disebutkan satu-persatu, telah membantu proses pengerjaan tugas akhir ini.

Penulis sadar bahwa Tugas Akhir ini belum sempurna dan masih banyak hal yang dapat diperbaiki. Saran, kritik dan masukan baik dari semua pihak sangat membantu penulis untuk pengembangan lebih lanjut. Penulis juga berharap supaya Tugas Akhir ini dapat menjadi aplikasi yang lebih bermanfaat.

Surabaya, 5 Juli 2017

Penulis

*Halaman ini sengaja dikosongkan*

## DAFTAR ISI

<b>LEMBAR PENGESAHAN .....</b>	<b>v</b>
<b>ABSTRAK .....</b>	<b>vii</b>
<b>ABSTRACT .....</b>	<b>ix</b>
<b>KATA PENGANTAR.....</b>	<b>xi</b>
<b>DAFTAR ISI.....</b>	<b>xiii</b>
<b>DAFTAR GAMBAR.....</b>	<b>xv</b>
<b>DAFTAR TABEL .....</b>	<b>xvii</b>
<b>BAB I PENDAHULUAN.....</b>	<b>1</b>
1.1. Latar Belakang .....	1
1.2. Perumusan Masalah.....	1
1.3. Batasan Masalah.....	2
1.4. Tujuan.....	2
1.5. Metodologi .....	2
1.6. Sistematika Penulisan .....	4
1.7. Relevansi .....	5
<b>BAB II TINJAUAN PUSTAKA DAN TEORI PENUNJANG .....</b>	<b>7</b>
2.1. Pengertian Robot .....	7
2.2. <i>Quanser Mobile Robot Qbot</i> [1].....	7
2.3. <i>QUARC (Quanser Real-Time Control)</i> [3].....	8
2.4. Perangkat Keras.....	10
2.5. Navigasi Pada Mobile Robot.....	15
2.5.1. Metode Garis .....	16
2.5.2. Metode Garis Dan Busur.....	17
2.6. Jaringan Saraf Tiruan .....	19
2.6.1. Arsitektur Jaringan Saraf Tiruan [5] [3].....	20
2.6.2. Fungsi Aktivasi <i>Neural Network</i> [3] .....	21
2.6.3. Proses <i>Learning Neural Network</i> [3].....	26
2.7. Kinematika <i>Mobile Robot Qbot</i> .....	29
<b>BAB III PERANCANGAN SISTEM.....</b>	<b>33</b>
3.1. Sistem Penerapan Navigasi Pada <i>Mobile Robot Qbot</i> .....	33
3.2. Desain <i>Kinematics</i> .....	34
3.2.1. Desain <i>Forward Kinematics</i> .....	34
3.2.2. Desain <i>Inverse Kinematics</i> .....	36
3.4. Rancangan Simulasi .....	37
3.5. Rancangan Navigasi <i>Mobile Robot</i> .....	40
3.6. Struktur Jaringan Saraf Tiruan [2].....	41

3.6.1. Perhitungan <i>Feedforward</i> .....	43
3.6.2. Perhitungan <i>Backward (Backpropagation)</i> .....	43
<b>BAB IV PENGUJIAN .....</b>	<b>54</b>
4.1. Implementasi Perencanaan Navigasi .....	55
4.2. Pengujian Identifikasi Obyek.....	55
4.3 Pengujian Pada Jalur.....	59
4.3.1 Pengujian Pada Jalan Lurus dan Robot Jalan Lurus .....	59
4.3.2 Penempatan Obyek Berada di Sisi Kanan dan Kiri Robot .....	60
4.3.3 Kondisi Navigasi Jalan Belok Kanan dan Robot Belok Kanan .....	60
4.3.4 Kondisi Navigasi Jalan Belok Kiri dan Robot Belok Kiri .....	61
4.3.5 Pengujian Pada Jalur .....	62
<b>BAB V KESIMPULAN DAN SARAN.....</b>	<b>64</b>
5.1 Kesimpulan .....	67
5.2 Saran .....	67
<b>DAFTAR PUSTAKA .....</b>	<b>69</b>
<b>LAMPIRAN.....</b>	<b>71</b>
<b>BIODATA PENULIS .....</b>	<b>120</b>



## DAFTAR GAMBAR

<b>Gambar 2.1</b>	Aoutonomus Mobile Robot Qbot.....	8
<b>Gambar 2.2</b>	Rangka Mobile Robot Qbot.....	11
<b>Gambar 2.3</b>	Tombol Pada <i>Mobile Robot Qbot</i> .....	12
<b>Gambar 2.4</b>	Kamera USB <i>Logitech Quickcam 9000</i> .....	12
<b>Gambar 2.5</b>	Baterai <i>Mobile Robot Qbot</i> .....	13
<b>Gambar 2.6</b>	Sensor InfraRed .....	13
<b>Gambar 2.7</b>	Sensor Sonar .....	14
<b>Gambar 2.8</b>	Gumstix .....	14
<b>Gambar 2.9</b>	<i>Incremental Optical Encoder</i> dengan LED dan Dua Fototransistor.....	15
<b>Gambar 2.10</b>	Navigasi Persamaan Garis .....	17
<b>Gambar 2.11</b>	Navigasi Garis Dan Busur .....	18
<b>Gambar 2.12</b>	Pengenalan pertigaan .....	19
<b>Gambar 2.14</b>	<i>Single Layer Perceptron</i> .....	20
<b>Gambar 2.15</b>	<i>Multiple layer perceptron</i> .....	20
<b>Gambar 2.16</b>	Letak fungsi aktivasi pada <i>neural network</i> .....	21
<b>Gambar 2.17</b>	Fungsi aktivasi <i>Hard Limit</i> .....	21
<b>Gambar 2.18</b>	Fungsi aktivasi <i>Threshold</i> .....	22
<b>Gambar 2.19</b>	Fungsi aktivasi Bipolar ( <i>Symetric Hard Limit</i> ).....	22
<b>Gambar 2.20</b>	Fungsi aktivasi Bipolar (dengan <i>threshold</i> ) .....	23
<b>Gambar 2.21</b>	Fungsi aktivasi <i>Linear</i> (Identitas) .....	23
<b>Gambar 2.22</b>	Fungsi aktivasi <i>Saturating Linear</i> .....	24
<b>Gambar 2.23</b>	Fungsi aktivasi <i>Symetric Saturating Linear</i> .....	24
<b>Gambar 2.24</b>	Fungsi aktivasi <i>Sigmoid Biner</i> .....	25
<b>Gambar 2.25</b>	Fungsi aktivasi <i>Sigmoid Bipolar</i> .....	25
<b>Gambar 2.26</b>	<i>Differential Kinematic of Mobile Robot Qbot</i> [4] .....	30
<b>Gambar 3.1</b>	Diagram Blok Sistem	33
<b>Gambar 3.2</b>	<i>Forward Kinematics Relative to ICC</i> .....	34
<b>Gambar 3.3</b>	Pengaturan Warna Pada <i>Color Path</i> .....	37
<b>Gambar 3.4</b>	Perancangan Pengambilan <i>Image</i> Hingga <i>Motion Planing</i> .....	38
<b>Gambar 3.5</b>	Rancangan Simulink Secara Keseluruhan .....	39
<b>Gambar 3.6</b>	Rancangan Simulink Untuk Koordinat Target.....	39
<b>Gambar 3.7</b>	Rancangan Simulink Untuk Color Tresh hold, Find Bolb hingga robot Target .....	40
<b>Gambar 3.8</b>	Perpindahan Posisi Robot Ke Posisi Target.....	40
<b>Gambar 3.9</b>	Struktur Jaringan Saraf Tiruan dalam satu node .....	42

<b>Gambar 3.10</b> Rancangan Jarinagn Saraf Tiruan Secara Keseluruhan .....	42
<b>Gambar 3.11</b> Rancangan Navigasi.....	46
<b>Gambar 3.12</b> (a) Gambar Asli (b) Pengambilan Biru Pada Warna RGB (c) Menampilkan Hanya Warna Biru.....	47
<b>Gambar 3.13</b> Proses Image Crop .....	47
<b>Gambar 3.14</b> Proses Penentuan Titik Central Crop Obyek.....	48
<b>Gambar 3.15</b> (a) adalah Bentuk Bundar Yang akan Di-Crop dan (b) Hasil Crop .....	48
<b>Gambar 3.16</b> Analisa Untuk Crop gambar segitiga ditapilkan seperti pada gambar di atas.....	49
<b>Gambar 4.1</b> Obyek Yang Menjadi Navigasi .....	55
<b>Gambar 4.2</b> Gambar Obyek ditampilkan dan <i>background</i> dihitamkan .....	56
<b>Gambar 4.3</b> Obyek Dalam Bentuk Data Gambar Yang Telah di- <i>Crop</i> .....	56
<b>Gambar 4. 4</b> Benda-Benda Yang Dijadikan Obyek Navigasi.....	57
<b>Gambar 4. 5</b> Pengujian Pada Jalan lurus dan Robot Jalan Lurus .....	59
<b>Gambar 4. 6</b> Kondisi Jalan Belok Kanan dan Robot Belok Kanan...	60
<b>Gambar 4. 7</b> Gambar Jalur Jika Dilihat dari Belakang Robot.....	61
<b>Gambar 4. 8</b> Tampilan trajectory pada saat belok kanan .....	61
<b>Gambar 4. 9</b> Kondisi Navigasi Belok Kiri dan Robot Belok Kiri.....	61
<b>Gambar 4. 10</b> Trajectory yang ditampilkan Saat Belok Kiri.....	62
<b>Gambar 4. 11</b> Area Yang Masuk ke Dalam Capture Picture .....	62

## DAFTAR TABEL

Tabel 2.1 Spesifikasi dan model parameter mobile robot Qbot.....	8
Tabel 2.2 <i>Feature Simulink Matlab</i> .....	9
Tabel 2.3 Daftar notasi proses learning jaringan saraf tiruan.....	27
Tabel 4.1 Hasil pengujian pada 3 objek yang berbeda masing-masing memiliki 7 karakteristik berbeda-beda.....	57
Tabel 4.2 Target training dan keluaran training.....	58

*Halaman ini sengaja dikosongkan*

# **BAB I**

## **PENDAHULUAN**

### **1.1. Latar Belakang**

Semakin berkembangnya ilmu pengetahuan dan teknologi mendorong manusia menciptakan inovasi kreatif, canggih dan tepat guna. Penerapan inovasi tersebut bertujuan untuk mempermudah pekerjaan dan permasalahan yang dianggap memperlama tercapainya solusi yang diharapkan. Penerapan tersebut ada diantaranya menggunakan robot, karena sebagian besar khusus sistem otomasi membutuhkan teknologi robotik yang dapat beroperasi diberbagai kendala. Sistem otomasi mampu menyelesaikan permasalahan tersebut dengan baik dikarenakan telah diterapkan perancangan metode yang sesuai berdasarkan kebutuhan. Pada robot kebutuhan yang diperlukan berupa sesor-sensor, motor listrik, kontroller dan agotihma untuk menyelesaikan berbagai macam problem.

Penerapan sistem otomasi pada *kendaraan* menggunakan kamera tanpa pengemudi pada kendaraan dalam mencapai tujuan disebut *Automatic guided vehicle* (AGV). Pada 1989 diperkenalkan oleh Eberhardt bahwa tujuan belum diketahui, penerapan sistem dengan *host* tanpa kabel dan semacamnya. Pada kesempatan ini penerapan tersebut diimplementasikan pada *prototype* mobile robot yang memiliki sistem navigasi.

Sistem navigasi yang diterapkan pada mobile robot menggunakan data gambar yang diperoleh dari kamera robot yang kemudian diimplementasikan secara *online* menggunakan *wireless connection* antara robot dan komputer. Navigasi menentukan arah tujuan robot yang nantinya dianalisa diberbagai kondisi seperti saat belok maupun di pertigaan jalan. Jalur menentukan kecepatan roda pada mobile robot karena kondisi tersebut.

### **1.2. Perumusan Masalah**

Rumusan masalah bagaimana cara mengoperasikan robot pada jalur yang ditentukan karena kecepatan motor menentukan posisi robot dan saat robot berpindah posisi maka data gambar yang diambil juga berbeda sehingga dibutuhkan kemampuan pendeteksian jalur di setiap beberapa aksi robot seperti pada saat robot bergerak. Robot harus dapat

bermanuver di jalur secara otomatis dengan baik maka dibutuhkan suatu sistem yang dapat menentukan posisi kendaraan relatif terhadap jalan dan kecepatan motor dan sudut kendali kendaraan sehingga kendaraan tidak ke luar dari jalur. Kemudian komunikasi robot dan komputer sangat penting karena setiap aksi robot, robot mengirimkan posisinya ke *host*, oleh karena itu butuh algoritma yang baik dalam penyelesaian masalah tersebut.

### **1.3. Batasan Masalah**

Batasan masalah terdapat dalam pembahasan dalam menyelesaikan tugas akhir ini sebagai berikut:

1. Penerapan sistem navigasi pada mobile robot sebagai *prototype* kendaraan.
2. Pada implementasi alat robot tidak diberikan kontrol sehingga membuat jalannya robot tidak stabil seperti kelebihan belok.

### **1.4. Tujuan**

Tujuan dari tugas akhir ini adalah menerapkan sistem navigasi otomatis pada *mobile robot* berbasis data gambar sehingga mobile robot dapat beroperasi pada lintasannya. Posisi robot merupakan titik koordinat yang menjadi parameter untuk mendapatkan nilai sudut dan dari sudut maka mencari jarak antara posisi robot yang sekarang ke posisi target robot.

### **1.5. Metodologi**

Metodologi yang digunakan dalam penyusunan tugas akhir ini antara lain:

#### **1. Studi literatur**

Dilakukan dengan mempelajari literature yang mempelajari dasar-dasar dinamika dan kinematika robot baik ketika robot bergerak lurus maupun ketika berbelok. Kemudian mempelajari metode *Neural Network* yang digunakan untuk menemukan titik-titik koordinat target *mobile robot*.

#### **2. Menentukan Parameter**

Penentuan parameter – parameter robot digunakan untuk merancang kinematika maju (*forward kinematics*). Fungsi perancang

kinematika maju ini adalah untuk menentukan posisi awal atau posisi tujuan *end effector* yang diinginkan. Dari posisi awal yang dicari maka akan digunakan sebagai permasalahan kinematika balik (*inverse kinematics*). Penentuan parameter pada *image* yaitu penskalaan data gambar yang diperoleh menggunakan kamera kemudian dibandingkan dengan penskalaan posisi robot atau koordinat robot. Setelah robot telah memperoleh hasil perbandingan skala data gambar dan posisi tersebut maka dilakukan pendeteksian kembali agar sistem tetap berjalan.

### **3. Perancangan Sistem Navigasi *Mobile Robot***

Hasil Penskalaan dari data gambar ke posisi dalam bentuk koordinat maka dilakukan perancangan sistem yang memiliki umpan balik dimana robot memperoleh data gambar berupa jalur dan robot bergerak mengikuti jalur menggunakan kinematika maju (*forward kinematics*) akan digunakan sebagai penentuan titik robot. Dari titik awal atau titik yang dituju akan dicari semua besar sudut sendi pada robot untuk mencapai titik tersebut menggunakan kinematika balik (*inverse kinematics*). *Mobile robot* diberi tugas menentukan posisinya yang nantinya akan dihadapkan ke jalur dan sebuah benda untuk diidentifikasi. Pengidentifikasian benda tersebut menggunakan jaringan saraf tiruan. Sehingga robot memiliki dua tugas yaitu proses penentuan posisi dan pengidentifikasian benda yang saling berhubungan seolah-olah robot memiliki penglihatan dan ingatan dalam penerapan sistem cerdas dalam menyelesaikan navigasi.

### **4. Simulasi Pergerakan *Mobile Robot***

Pada tahap simulasi *mobile robot* adalah sistem yang diberikan pada robot untuk mendeteksi jalur yang akan dilalui menggunakan titik koordinat pada jalur yang telah terdeteksi. Robot mampu menyeleksi titik koordinat pada jalur dimulai dari pemrosesan dan penyeleksian data dari titik terdekat dan mengetahui koordinat jalur pada piksel gambar. Jika koordinat lebih kekanan dari titik tengah maka *mobile robot* akan menuju ke kanan, dan sebaliknya jika koordinat yang dideteksi lebih ke kiri maka robot akan bergerak ke kiri. Namun jika koordinat yang dideteksi yang merupakan jalur *mobile robot* maka robot jalan lurus dan selalu menyesuaikan koordinat target yang akan dideteksi robot sebagai jalur.

## 5. Penulisan Buku Tugas Akhir

Penulisan buku tugas akhir ditulis dengan baik dan sesuai prosedur. Buku terdiri dari 5 bab inti yaitu, pendahuluan berisikan uraian singkat tugas akhir, dasar teori berisikan materi – materi pengerjaan tugas akhir, perancangan metode untuk sistem navigasi, hasil analisa dan penutup.

### 1.6. Sistematika Penulisan

Pembahasan pada Tugas Akhir ini dibagi menjadi lima bab dengan sistematika penulisan sebagai berikut :

#### **BAB I : Pendahuluan**

Bab ini mengenai penjelasan latar belakang, rumusan masalah, batasan masalah, tujuan penelitian, sistematika penulisan, serta relevansi.

#### **BAB II : Teori Dasar**

Bab ini menjelaskan tinjauan pustaka, konsep dasar dari robot, *forward kinematic*, *inverse kinematics* dan *jaringan saraf tiruan* yang dibutuhkan pada penerapan navigasi mobile robot.

#### **BAB III : Perancangan Sistem**

Bagian ini berisi pembahasan tentang cara menentukan parameter *mobile robot Qbot*. Mencari persamaan *forward kinematics* untuk mendapatkan posisi yang diinginkan. Pembentukan struktur *neural network* untuk mengenali pola benda atau warna benda. Perancangan ini diperlukan *mobile robot* bergerak dari x,y robot ke titik x,y target.

#### **BAB IV : Pengujian dan Analisis Sistem**

Bab ini mengimplementasikan hasil simulasi dan pengaplikasian *navigasi mobile robot* serta analisis data pada setiap pengujian.



## **BAB V : Penutup**

Pada Bab IV diambil suatu kesimpulan dan saran yang baik sebagai bahan evaluasi penelitian saat ini dan selanjutnya.

### **1.7. Relevansi**

Hasil yang diharapkan dari pembuatan tugas akhir ini adalah dapat diaplikasikan di kehidupan sehari-hari hingga terciptanya kendaraan cerdas dan juga membantu meringankan pekerjaan manusia. Pengembangan lebih lanjut diharapkan menjadi referensi dalam pengembangan sistem navigasi mobile robot.

*Halaman ini sengaja dikosongkan*

## **BAB II**

### **TINJAUAN PUSTAKA DAN TEORI PENUNJANG**

Sistem Pengenalan pola pada *mobile robot* secara umum mempelajari data yang diperoleh dari kamera kemudian diproses yang dikenal sebagai *image processing*. Robot diberi tugas menentukan posisinya yang nantinya akan dihadapkan ke jalur dan sebuah benda untuk diidentifikasi. Pengidentifikasian benda tersebut menggunakan jaringan saraf tiruan. Sehingga robot memiliki dua tugas yaitu proses penentuan posisi dan pengidentifikasian benda yang saling berhubungan seolah-olah robot memiliki penglihatan dan ingatan dalam penerapan sistem cerdas dalam menyelesaikan navigasi.

#### **2.1. Pengertian Robot**

Robot pertama kali diperkenalkan kedalam kosa kata oleh dramawan Ceko Karel Capek (1920) yaitu "*robota*", dalam bahasa Ceko memiliki arti "bekerja". Sejak saat itu istilah tersebut telah diaplikasikan untuk bermacam – macam perangkat mekanik, seperti teleoperator, kendaraan bawah air, dan lain sebagainya. Hampir semua yang beroperasi di bawah kendali dari komputer dapat disebut dengan robot. RIA (*Robot Institute of America*) mendefinisikan bahwa sebuah robot merupakan manipulator multifungsi bisa di program (*reprogrammable*) ulang yang didesain untuk memindahkan material, suku cadang, peralatan atau perangkat khusus dengan variabel gerakan yang diprogram untuk berbagai tugas

#### **2.2. Quanser Mobile Robot Qbot [1]**

*Quanser mobile robot merupakan autonomus robot* buatan *Irobot* yang telah dilengkapi sistem kontrol terpusat pada sebuah komputer. *Mobile robot* ini diberi nama singkat yaitu robot Qbot. Robot ini dirancang untuk menguasai ground yang diintegrasikan dengan perangkat keras seperti sensor kamera, sensor *infra red*, sensor *sonar*, sensor posisi dan sensor jarak. Kemampuan robot mengirimkan data pada komputer sebagai *host*-nya menggunakan perangkat lunak *QUARC* (*Quanser Controller Module*). *QUARC* merupakan *software quanser* pada komputer. *Software* ini memfasilitasi pemrograman menggunakan *MATLAB* sehingga bisa dikembangkan menggunakan *Simulink* dan *M.File* sesuai kebutuhan pengguna. Komunikasi robot dengan robot

menggunakan *Gumstix* yang memberikan sinyal dari komputer ke robot dan sebaliknya. Bentuk *Quanser Mobile Robot Qbot* seperti Gambar 2.1.



**Gambar 2.1** Aoutonomus Mobile Robot *Qbot*

Spesifikasi dan model parameter *mobile robot Qbot* dapat dilihat pada Tabel 2.1.

**Tabel 2.1** Spesifikasi dan model parameter *mobile robot Qbot*

No	Diskripsi	Simbol	Value	Satuan
1	Diameter mobile robot Qbot	d	0.34	m
2	Tinggi mobile robot Qbot (dengan tambahan kamera)	h	0.19	m
3	Kecepatan maksimum dari Qbot mobile robot	vmax	0.5	m/s
4	Berat Qbot <i>mobile robot</i>	m	2.92	kg

### 2.3. QUARC (Quanser Real-Time Control) [3]

*QUARC* menyediakan akses melalui *Simulink Matlab* dalam pembuatan dan pengembangan kontrol serta pengujian pada *mobile robot Qbot*. Pada *Simulink Matlab* dilakukan perancangan model hingga pengontrolan dan kemudian *build* dan langsung diunduh serta dijalankan pada target secara *real time*. *Gimustix* berperan pada proses ini yang secara lansung melakukan komunikasi dari komputer sebagai *host* dan *mobile robot Qbot* sebagai target yang akan dikontrol dan mendapatkan respon dari *variable* robot yang sifatnya berganti-ganti. Proses unduhan ini dilakukan secura *wireless* ke robot.

Pada *Simulink Matlab* ada beberapa *fiture* yang telah disediakan untuk digunakan dalam pengontrolan *mobile robot* seperti pada Tabel 2.2.

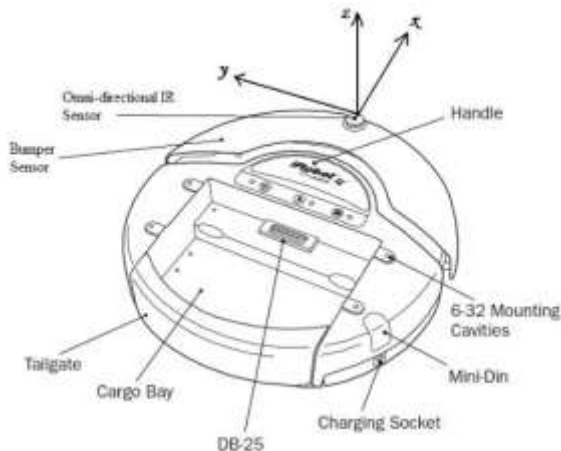
**Tabel 2.2** *Feature Simulink Matlab*

No	Set Blok	Deskripsi
1	Aplikasi	Set blok ini berfungsi untuk menerapkan algoritma navigasi dengan menggunakan data encoder roda dan bump sensors untuk navigasi dan menghindari rintangan.
2	<i>Image Processing</i>	Set blok untuk pengolahan data gambar menggunakan <i>Open CV</i> . Perangkat kamera terhubung USB ke <i>mobile robot</i> data kemudian dikirim ke komputer secara <i>wireless</i> dengan menambahkan menggunakan <i>toolbox Vision Image Get IPlimage Data</i> .
3	<i>Romba Drive</i>	Blok <i>Block QUARC Input</i> untuk pergerakan robot. Terdapat dua <i>input</i> , yaitu Pertama kecepatan untuk roda kanan dan roda kiri bisa diatur dan kedua adalah radius, untuk mengetahui radius dari pergerakan lurus maka <i>input</i> untuk radius 32768 atau 32767.
4	<i>Romba Query List</i>	Blok ini merupakan <i>Block QUARC</i> yang tujuannya untuk akses data sensor Qbot. Input <i>pack_id</i> berkisar 7-42, dan setiap input akan menampilkan nilai dari output pack. <i>Pack ID Value:</i> 8 = Dinding (0 = tidak ada dinding, 1 = dinding terlihat) 19 = Jarak (Jarak yang Qbot telah melakukan perjalanan dalam milimeter) 20 = Sudut (dalam derajat <i>mobile robot Qbot</i> )

		<p>22 = Tegangan (Tegangan baterai milivolt)</p> <p>23 = Arus (Arus dalam miliampere mengalir dari baterai)</p> <p>Ke prosesor <i>mobile robot Qbot</i>)</p>
5	<i>Romba Initilize</i>	<p>Blok ini merupakan <i>Block QUARC</i> yang tujuannya untuk sambungan serial komputer ke <i>Qbot</i> .</p> <p>Identifikasi melakukan komunikasi ini maka menggunakan URI (<i>Universal Resource Identifier</i>).  serial://localhost:1baud=57600,word=8, parity=none,stop=1.</p> <p>Target terhubung ke port serial Qbot dengan parameter yang sesuai ketentuan seperti URI di atas sehingga mampu melakukan komunikasi dengan komputer.</p>
6	<i>Hil Initiallize Hil-1(q8-0)</i>	<p>Blok ini merupakan <i>Block QUARC</i> yang tujuannya untuk inisialisasi <i>model</i>. Modul pada <i>QUARC</i> menggunakan <i>HillCard</i>. <i>HIL Read Write</i> memiliki prosedur harus memiliki referensi blok untuk menentukan <i>board</i> yang akan dimasukan ke dalam blok.</p>
7	<i>Hil Read Write</i>	<p>Blok bertugas untuk membaca saluran analog yang berupa sensor infra red dan sonar dimana untuk pengaturan analognya menggunakan <i>PWM</i> yang menyediakan operasi <i>write</i>.</p>

## 2.4. Perangkat Keras

Perangkat keras pada *Mobile Robot Qbot* merupakan semua aspek yang terlihat dan berbentuk dimensi 3. Komponen keras ini menunjang kerja robot sehingga robot mampu beroperasi dengan saling melengkapi dan terstruktur.



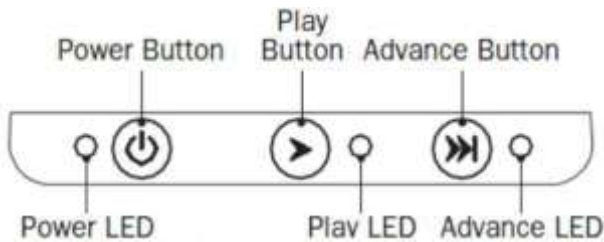
**Gambar 2.2** Rangka *Mobile Robot Qbot*

Adapun perangkat keras *Mobile robot Qbot* adalah sebagai berikut.

### **1. Tombol *Mobile Robot Qbot***

Tombol pada *mobile robot* seperti yang terdapat pada Gambar 2.4 terdapat empat tombol digunakan untuk pengoperasian robot. Tombol pertama dari sebelah kiri pada piranti robot yaitu tombol *on* atau *off Qbot mobile robot Qbot*. Tombol *On* atau *Off* dilengkapi *Led Indicator* disebelah kiri tombol *Power* robot. *Indicator* berupa lampu merah kedip-kedip sebagai baterai habis. Kemudian Jika *indicator* kuning merupakan baterai terisi penuh. Tombol tengah adalah tombol *Play Button*. Fungsi *Play Button* berfungsi untuk pengoperasian *mobile robot Qbot*. Tombol selanjutnya adalah *Advance Button* terdapat di sebelah kanan yang berguna untuk menjalankan demo untuk *atounomous mobile robot Qbot*. Dan sama seperti tombol *Power* masing-masing tombol memiliki *LED Indicator* yang mendakan bahwa robot menjalankan perintah dari tombol yang sudah tersedia pada *Qbot mobile robot*.

Di bawah ini pada Gambar 2.3 adalah penggambaran tombol yang terdapat pada *Qbot mobile robot*.



**Gambar 2.3** Tombol Pada *Mobile Robot Qbot*

## 2. Kamera USB *Logitech Quickcam Pro 9000*

Kamera yang digunakan pada *mobile robot Qbot* adalah kamera USB *Logitech Quickcam Pro 9000*. Kamera ini terpasang langsung pada *mobile robot* dengan kabel USB. Kamera digunakan untuk pengambilan data gambar pada *robot vision* sehingga dapat mengambil gambar dan ditampilkan pada layar komputer. Data gambar dari robot bisa di akuisisi dan diproses lagi untuk dijadikan kondisi yang diinginkan pengguna baik untuk analisis maupun untuk pergerakan robot. Bentuk fisik dari kamera USB *Logitech Quickcam Pro 9000* seperti pada Gambar 2.4.



**Gambar 2.4** Kamera USB *Logitech Quickcam 9000*

## 3. Baterai *Advance Power System (APS)*

Baterai *Advance Power System (APS)* merupakan baterai yang digunakan pada *mobile robot Qbot*. Baterai ini terletak di tengah bagian bawah *mobile robot Qbot*. Penggunaan baterai saat beroperasi mampu berlangsung sekitar dua jam setelah terisi penuh. Pengisian baterai saat kapasitas minimum ke maksimum membutuhkan waktu 3 jam lebih lama. Bentuk fisik dari Baterai *Advance Power System* seperti pada Gambar 2.5





**Gambar 2.5** Baterai *Mobile Robot Qbot*

Lampu daya Qbot mengindikasikan tingkat daya baterai. Lampu hijau menandakan baterai masih terisi penuh, kemudian secara bertahap berubah menjadi merah bila baterai akan habis. Baterai memerlukan waktu kurang dari tiga jam untuk pengisian baterai. Ketika pengisian, lampu daya akan berkedip perlahan dengan warna jingga dan akan berubah menjadi hijau ketika terisi penuh.

### **3. Sensor Sonar**

Sensor Sonar SHARP 2Y0A02 merupakan sensor inframerah dengan jarak 20-150 cm. Terdapat lima buah sensor inframerah yang terpasang pada Qbot. Sensor terhubung ke saluran *input analog* dari Qbot DAC, yang kemudian dapat dibaca dengan menggunakan blok HIL *Read Write*. Bentuk fisik dari sensor sonar seperti pada Gambar 2.6



**Gambar 2.6** Sensor InfraRed

### **4. Sensor sonar MaxSonar-EZ0**

Sensor sonar MaxSonar-EZ0 mendeteksi obyek dari 0 *inchi* sampai 254 *inchi* dengan resolusi 1 *inchi*. Obyek antara 0 *inchi* dan 6 *inchi* berkisar sebagai 6 *inchi*. Terdapat tiga sensor pada Qbot. Sensor yang terhubung ke saluran masukan lain dari Qbot DAC, yang

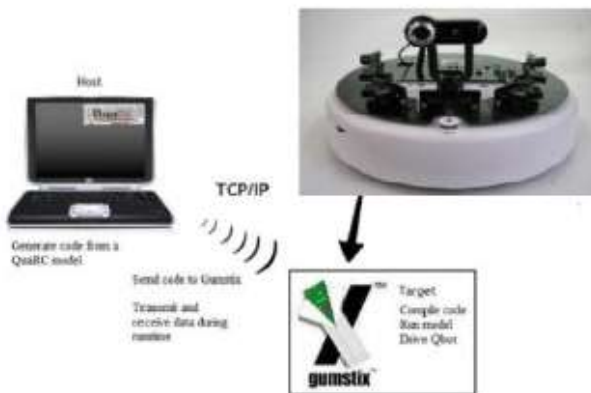
kemudian dapat dibaca dengan menggunakan blok HIL *Read Write*. Gambar 2.7 menunjukkan sensor sonar MaxSonar-EZ0.



**Gambar 2.7** Sensor Sonar

## 5. *Gumstix*

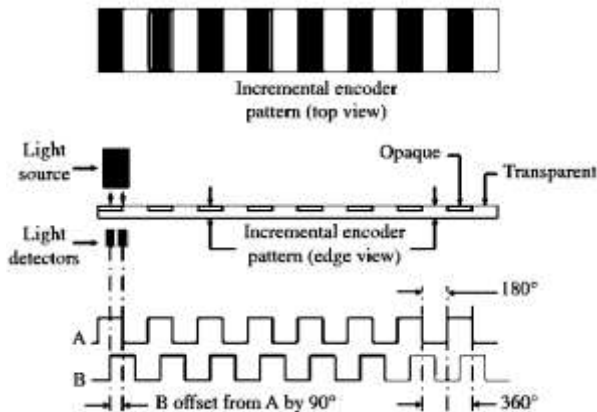
*Gumstix* pada *Qbot* robot yang berfungsi sebagai penghubung atau perantara agar Simulink MATLAB secara langsung dapat diunduh ke *mobile robot Qbot*. Kompilasi dan unduhan program pada robot berhasil melalui ekstensi perangkat lunak yaitu QUARC. *Gumstix* dihubungkan langsung pada *mobile robot Qbot* yang nantinya akan memancarkan jaringan *wifi* ke komputer. *Wifi* menyediakan koneksi nirkabel antara target *Gumstix* dan komputer *host*. Gambar 2.8 merupakan *gumstix* pada *mobile robot*.



**Gambar 2.8** Gumstix

## 6. Rotary Encoder

Rotary Encoder merupakan sensor pendeteksi posisi dan kecepatan dengan mengubah perpindahan mekanik menjadi sinyal elektrik. Penggunaan *rotary encoder* bertujuan untuk mengetahui perpindahan linier. Nilai perpindahan diukur dan dikomunikasikan oleh encoder dan sinyal keluaran dari *rotary encoder* berupa sinyal digital. Terdapat dua jenis *encoder* yaitu *encoder incremental* dan *encoder absolute* yang mencakup beberapa teknik pendeteksian, salah satunya adalah jenis pendeteksian menggunakan optik. *Encoder* optik merupakan komponen optoelektronik *transmitter/receiver*, *transmitter* berupa sumber cahaya, biasanya berupa satu atau lebih LED dan *receiver* biasanya terdiri dari dua atau lebih fototransistor. Berikut Gambar 2.9 *encoder* optik.



**Gambar 2.9** *Incremental Optical Encoder* dengan LED dan Dua Fototransistor.

## 2.5. Navigasi Pada Mobile Robot

Navigasi merupakan penentuan posisi dan arah terhadap tujuan dalam suatu pemetaan. Penentuan posisi *mobile robot* berubah sehingga mempengaruhi ingatan yang baik agar tidak membingungkan. Pada kehidupan sehari-hari kendaraan mobil dikendalikan oleh manusia, manusia mempelajari navigasi dari apa yang dilihat seperti arah, batas jalan dari kendaraan, sehingga dapat diketahui posisi kendaraan.

Pencarian koordinat pada pertigaan atau perempatan adalah metode yang digunakan dalam navigasi robot saat ada kondisi, namun jika tidak ditemui adanya pertigaan maka robot akan berjalan seperti biasa yaitu mengikuti trek yang telah ditentukan yaitu menuju koordinat terdekat. Untuk mengikuti garis tersebut dibutuhkan prinsip kerja dari navigasi kendaraan secara otomatis adalah dengan mencari batas tepi kanan dan batas tepi kiri jalan dari kendaraan, sehingga dapat diketahui posisi relatif kendaraan terhadap jalur dan robot diposisikan pada tengah jalur. Navigasi kendaraan ada beberapa metode seperti metode garis, metode busur, dan metode garis dan busur.

### 2.5.1 Metode Garis

Metode garis untuk navigasi *Qbot mobile robot* merupakan metode yang mencari batas tepi kanan, batas tepi kiri, dan batas tepi depan jalan. Menggunakan deteksi garis dengan mencari sudut tertentu terhadap posisi kamera, seperti pada Gambar 2.8. Batas tepi depan jalan menggunakan sudut  $\theta$  sebesar  $90^0$ . Batas tepi depan jalan digunakan sebagai kontrol kecepatan kendaraan. Langkah-langkah yang dilakukan menggunakan metode garis adalah :

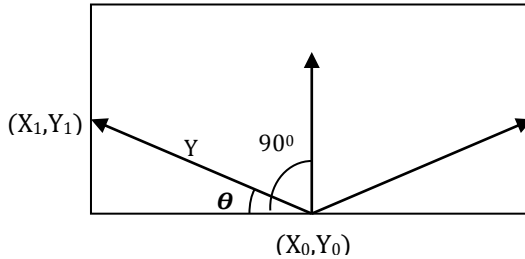
1. Menentukan titik awal  $X_0$  dan  $Y_0$  terlebih dahulu
2. Menentukan nilai  $Y_1$  berdasarkan sudut  $\theta$  yang diinginkan
 
$$Y_1 = X_0 \tan \theta \quad (2.1)$$
3. Setelah  $Y_1$  diketahui, maka menentukan panjang garis horisontal yang diinginkan  $X_1$  sehingga didapatkan nilai gradien  $m$

$$m = \frac{Y_0 - Y_1}{X_0 - X_1} \quad (2.2)$$

4. Setelah titik awal ( $X_0$  dan  $Y_0$  pada langkah 1) dan garis horisontal yang diinginkan ( $X_1$ ) ditentukan serta gradien ( $m$ ) diketahui, maka dimasukkan dalam rumus

$$Y = X_0 - m (X_0 - X_1) \quad (2.3)$$

5. Sepanjang  $Y$  dicari batas jalan sehingga didapatkan panjang jarak antara kendaraan dengan batas jalan.



**Gambar 2.10** Navigasi Persamaan Garis

### 2.5.2 Metode Garis Dan Busur

Metode busur pada navigasi *mobile robot* batas tepi jalur menggunakan garis busur dengan lengkungan tertentu, seperti pada Gambar 2.11. Pada metode busur mempunyai dua cara yang berbeda untuk mengatur sudut kendali dengan mengatur kecepatan. Mengatur kecepatan kendaraan pada metode busur adalah dengan menggunakan selisih batas tepi kanan dan batas depan kiri.

Langkah-langkah metode busur untuk mengatur sudut kendali adalah :

1. Menentukan titik pusat lingkaran  $Y_1$  berdasarkan batas tepi kiri, batas tepi kanan, dan batas depan jalan yang diinginkan.

$$r_1(Y_{21} - Y_1)^2 + (Y_{21} - X_1)^2 \quad (2.4)$$

$$r_2(Y_{21} - Y_1)^2 + (Y_{22} - X_1)^2 \quad (2.5)$$

Keterangan :

1.  $r_1$  = jari-jari yang pertama
2.  $r_2$  = jari-jari yang kedua
3.  $Y_{21}$  = nilai Y yang pertama
4.  $X_{21}$  = nilai X yang pertama
5.  $Y_{22}$  = nilai Y yang kedua
6.  $X_{22}$  = nilai X yang kedua
7.  $Y_1$  = nilai Y pusat lingkaran
8.  $X_1$  = nilai X pusat lingkaran

$$r_1 = r_2, \text{ maka didapatkan nilai } Y_1 \quad (2.6)$$

2. Penentuan jari-jari lingkaran ( $r$ ). dilihat dari titik pusat lingkaran dan salah satu batas tepi maka didapatkan jari-jari.

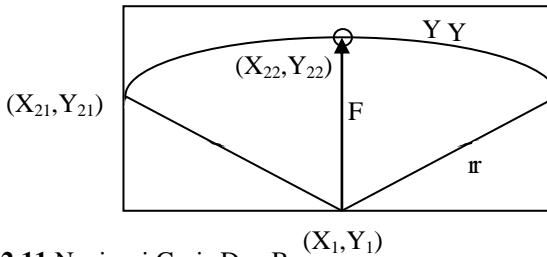
$$r = \sqrt{(Y_{21} - Y_1)^2 + (Y_{22} - X_1)^2} \quad (2.7)$$

3. Setelah diketahui pusat lingkaran ( $X_1, Y_1$ ), jari-jari lingkaran ( $r$ ), dan titik horisontal yang diinginkan ( $X$ ). Setelah itu dapat diketahui titik-titik dari garis busur ( $X, Y$ ) yang membentuk lengkungan.

$$r = Y_1 \sqrt{r^2 + (X_{11} - X_1)^2} \quad (2.8)$$

4. Sepanjang  $Y$  digunakan untuk mendeteksi batas tepi kiri dan batas tepi kanan dari tengah kesamping, sehingga didapatkan panjang jarak batas tepi kiri dan batas tepi kanan.

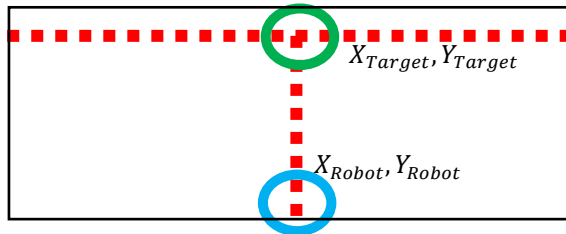
Langkah-langkah metode busur untuk mengatur kecepatan dengan menentukan garis tengah dan mencari batas depan dengan garis yang tegak lurus terhadap garis tengah, sehingga diketahui panjang jarak garis depan dengan kendaraan ( $F$ ).



**Gambar 2.11** Navigasi Garis Dan Busur

### 2.5.3 Pendeteksian Pertigaan

Pengenalan pertigaan pada mobie robot Qbot menggunakan analisa piksel dalam bentuk matrik seperti pada Gambar 2.12.



**Gambar 2.12** Pengenalan pertigaan

Keterangan:

1.  $X_{Robot}$  adalah koordinat X robot saat sekarang
2.  $Y_{Robot}$  koordinat Y robot saat sekarang
3.  $X_{Target}$  adalah koordinat X yang akan dituju oleh robot pada pertigaan
4.  $Y_{Robo}$  adalah koordinat Y yang akan dituju oleh robot pada pertigaan

## 2.6. Jaringan Saraf Tiruan

Jaringan saraf tiruan merupakan representasi dari otak buatan yang tujuannya adalah mensimulasikan proses pembelajaran pada sebuah obyek layaknya meniru kemampuan otak manusia bekerja. Jaringan saraf tiruan dapat disimulasikan dan juga diimplementasikan pada sebuah robot yang terprogram. Tugas utama dari algoritma pemrograman pada jaringan saraf adalah menyelesaikan sejumlah proses perhitungan selama proses pembelajaran. *Neural Network* dikembangkan sebagai turunan model matematika dari kesadaran sistem saraf biologis didasarkan pada:

1. Pemrosesan informasi terjadi pada *neuron*.
2. Sinyal lewat di antara *neuron* menciptakan jaringan koneksi.
3. Setiap koneksi penghubung memiliki bobot
4. *Neuron* mempunyai fungsi aktivasi pada jaringan *input*-nya untuk menentukan sinyal *output*-nya.

Karakteristik jaringan saraf tiruan sebagai berikut :

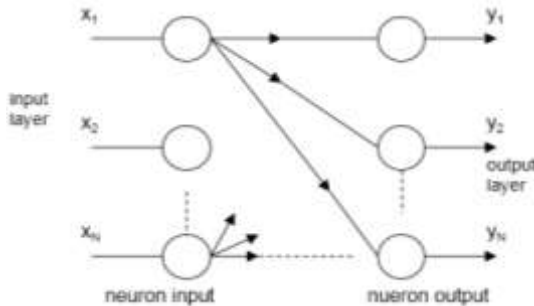
1. Hubungan antar *neuron* disebut arsitektur *neural network*.
2. Adanya proses latihan, pembelajaran, atau Algoritma.
3. Terdapat fungsi aktivasi.

### 2.6.1 Arsitektur Jaringan Saraf Tiruan [5] [3]

Susunan *neuron* disebut arsitektur jaringan saraf tiruan. Terdapat tiga jenis arsitektur jaringan saraf tiruan berdasarkan lapisannya, yaitu :

#### 1. Arsitektur Jaringan *single layer*

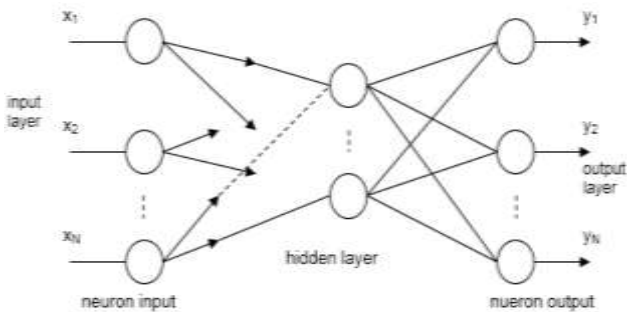
Jaringan *single-layer* merupakan jaringan *Feedforward*, karakteristik susunan arsitekturnya terdapat sinyal *input* mengalir ke *ouput*.



**Gambar 2.13** *Single Layer Perceptron*

#### 2. Arsitektur Jaringan *Multi layer*

Jaringan *multi layer* adalah jaringan yang mempunyai lebih dari satu lapisan *nodes* (titik hubung / koneksi) di antara input dan outputnya. Biasanya ada lapisan untuk menyatakan berat (*weight*) di antara dua level yang berdekatan. Jaringan *multi layer* dapat menyelesaikan permasalahan yang lebih kompleks. Contoh jaringan *multi layer* dapat dilihat pada Gambar 2.15.

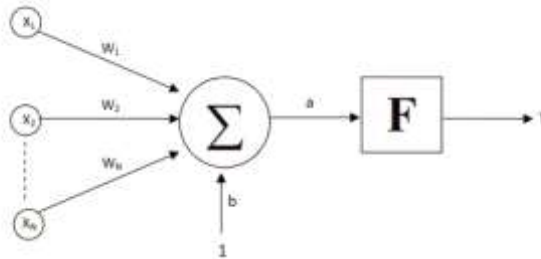


**Gambar 2.14** *Multiple layer perceptron*



### 2.6.2 Fungsi Aktivasi *Neural Network* [3]

Aktifasi *neural network* artinya yaitu fungsi pengaktif *neuron* pada jaringan saraf. Banyak fungsi yang dapat dipakai sebagai aktifasi, seperti fungsi unit step, impulse, sigmoid, dan lain sebagainya. Letak fungsi aktivasi dalam struktur *neural network* dapat dilihat pada Gambar 2.16.



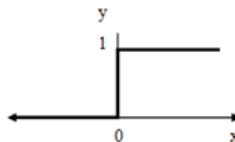
**Gambar 2.15** Letak fungsi aktivasi pada *neural network*

Ada beberapa fungsi aktivasi yang sering digunakan dalam *neural network*, antara lain:

1. Fungsi *Hard Limit*

Fungsi *hard limit* merupakan jaringan lapisan tunggal yang menggunakan *step function* untuk mengkonversikan *input* dari suatu variabel yang bernilai kontinyu ke suatu *output* biner (0 atau 1). Berikut Gambar 2.17 fungsi aktivasi *hard limit*.

$$y = \begin{cases} 0, & \text{jika } x \leq 0 \\ 1, & \text{jika } x > 0 \end{cases}$$



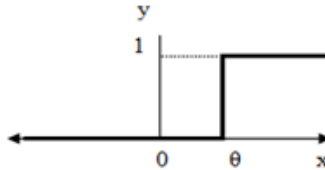
**Gambar 2.16** Fungsi aktivasi *Hard Limit*

2. Fungsi *Threshold*

Fungsi *threshold* dengan menggunakan nilai ambang sering juga disebut dengan nama fungsi *threshold* atau fungsi *heaviside*.

Fungsi *threshold* (dengan nilai ambang 0) dirumuskan sebagai berikut dan gambarnya dapat dilihat pada Gambar 2.18.

$$y = \begin{cases} 0, & \text{jika } x < \theta \\ 1, & \text{jika } x \geq \theta \end{cases}$$

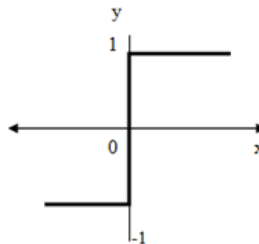


**Gambar 2.17** Fungsi aktivasi *Threshold*

3. Fungsi Bipolar atau *Symetric Hard Limit*

Fungsi bipolar dengan *symetric hard limit* sebenarnya hampir sama dengan fungsi undak biner, hanya saja output yang dihasilkan berupa 1, 0, -1. Fungsi *symetric hard limit* dirumuskan sebagai berikut dan gambarnya dapat dilihat pada Gambar 2.19.

$$y = \begin{cases} 1, & \text{jika } x > 0 \\ 0, & \text{jika } x = 0 \\ -1, & \text{jika } x < 0 \end{cases}$$



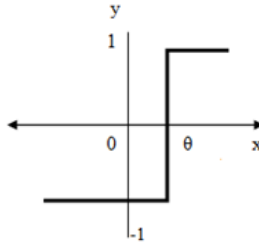
**Gambar 2.18** Fungsi aktivasi Bipolar (*Symetric Hard Limit*)

4. Fungsi Bipolar dengan *threshold*

Fungsi bipolar dengan *threshold* sebenarnya hampir sama dengan fungsi undak biner, hanya saja output yang dihasilkan berupa 1, 0 atau -1. Fungsi bipolar dengan nilai ambang 0 dirumuskan sebagai berikut dan gambarnya dapat dilihat pada

Gambar 2.20 hanya saja *output* yang dihasilkan berupa 1, 0, atau – 1. Fungsi ini dirumuskan sebagai berikut:

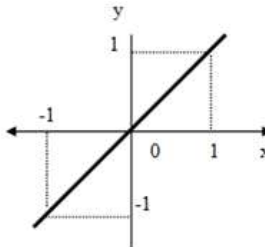
$$y = \begin{cases} 1, & \text{jika } x > \theta \\ 0, & \text{jika } x = \theta \\ -1, & \text{jika } x < \theta \end{cases}$$



**Gambar 2.19** Fungsi aktivasi Bipolar (dengan *threshold*)

5. Fungsi *Linear* (Identitas)

Fungsi ini memiliki nilai *output* yang sama dengan nilai inputnya, dapat dilihat pada Gambar 2.21 dan dapat dirumuskan sebagai berikut: dimana  $Y = X$

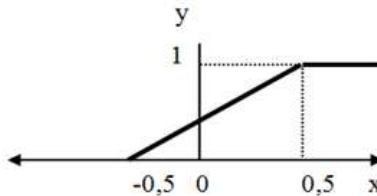


**Gambar 2. 20** Fungsi aktivasi *Linear* (Identitas)

6. Fungsi *Saturating Linear*

Fungsi ini akan bernilai 0 jika inputnya kurang dari  $-\frac{1}{2}$ , dan akan bernilai 1 jika inputnya lebih dari  $\frac{1}{2}$ . Sedangkan jika nilai input terletak antara  $-\frac{1}{2}$  dan  $\frac{1}{2}$ , maka *output*nya akan bernilai sama dengan nilai inputnya ditambah  $\frac{1}{2}$ . Fungsi *saturating linear* ditunjukkan pada Gambar 2.22 dan dirumuskan sebagai berikut:

$$y = \begin{cases} 1; & \text{jika } x \geq 0,5 \\ x + 0,5; & \text{jika } -0,5 \leq x \leq 0,5 \\ 0; & \text{jika } x \leq -0,5 \end{cases}$$

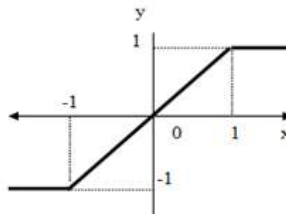


**Gambar 2.21** Fungsi aktivasi *Saturating Linear*

7. Fungsi *Symetric Saturating Linear*

Fungsi ini akan bernilai  $-1$  jika inputnya kurang dari  $-1$ , dan akan bernilai  $1$  jika inputnya lebih dari  $1$ . Sedangkan jika nilai input terletak antara  $-1$  dan  $1$ , maka outputnya akan bernilai sama dengan nilai inputnya. Fungsi *symetric saturating linear* ditunjukkan pada Gambar 2.23.

$$y = \begin{cases} 1; & \text{jika } x \geq 1 \\ x; & \text{jika } -1 \leq x \leq 1 \\ -1; & \text{jika } x \leq -1 \end{cases}$$



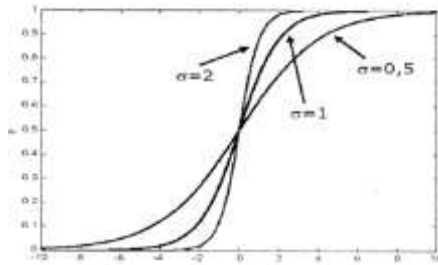
**Gambar 2. 22** Fungsi aktivasi *Symetric Saturating Linear*

8. Fungsi *Sigmoid Biner*

Dengan menggunakan metode *backpropagation*, mempunyai range  $0$  sampai  $1$ . Biasanya digunakan untuk jaringan saraf yang membutuhkan nilai output yang terletak pada interval  $0$  sampai dengan  $1$ , juga pada jaringan saraf yang nilai outputnya  $0$  atau  $1$ . Fungsi ini dirumuskan sebagai berikut:

$$y = f(x) = \frac{1}{1 + e^{-\sigma x}}$$

$$\text{dengan : } f'(x) = \sigma f(x) [1 - f(x)]$$



**Gambar 2. 23** Fungsi aktivasi *Sigmoid Biner*

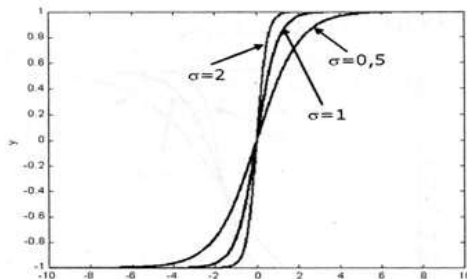
#### 9. Fungsi *Sigmoid Bipolar*

Fungsi ini hampir sama dengan fungsi *sigmoid biner*, tetapi output fungsi ini memiliki range 1 sampai  $-1$ . Fungsi ini dirumuskan sebagai berikut:

$$y = f(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$$

$$\text{atau } y = f(x) = \frac{1 - e^{-2x}}{1 + e^{-2x}}$$

$$\text{dengan : } f'(x) = [1 + f(x)][1 - f(x)]$$



**Gambar 2. 24** Fungsi aktivasi *Sigmoid Bipolar*

### 2.6.3 Proses Learning Neural Network [3]

Algoritma pelatihan Backpropagasi (*Backpropagation*) atau ada yang menerjemahkannya menjadi propagasi balik, pertama kali dirumuskan oleh Werbos dan dipopulerkan oleh Rumelhart dan McClelland untuk dipakai pada JST, dan selanjutnya algoritma ini biasa disingkat dengan BP. Algoritma ini termasuk metoda pelatihan supervised dan didesain untuk operasi pada jaringan feed forward multi lapis.

Metoda BP ini banyak diaplikasikan secara luas. Sekitar 90 %, bahkan lebih BP telah berhasil diaplikasikan di berbagai bidang, diantaranya diterapkan di bidang finansial, pengenalan pola tulisan tangan, pengenalan pola suara, sistem kendali, pengolah citra medika dan masih banyak lagi keberhasilan BP sebagai salah satu metoda komputasi yang handal.

Algoritma ini juga banyak dipakai pada aplikasi pengaturan karena proses pelatihannya didasarkan pada hubungan yang sederhana, yaitu : Jika *output* memberikan hasil yang salah, maka pembobot (Weight) dikoreksi supaya erornya dapat diperkecil dan respon jaringan selanjutnya diharapkan akan lebih mendekati harga yang benar. BP juga berkemampuan untuk memperbaiki pembobot pada lapisan tersembunyi (*hidden layer*).

Secara garis besar, mengapa algoritma ini disebut sebagai propagasi balik, dapat dideskripsikan sebagai berikut: Ketika Jaringan diberikan pola *input* sebagai pola pelatihan maka pola tersebut menuju ke unit-unit pada lapisan tersembunyi untuk diteruskan ke unit-unit lapisan *output*. Kemudian unit-unit lapisan *output* memberikan tanggapan yang disebut sebagai *output* jaringan. Saat *output* jaringan tidak sama dengan *output* yang diharapkan maka *output* akan menyebar mundur (*backward*) pada lapisan tersembunyi diteruskan ke unit pada lapisan *input*. Oleh karenanya maka mekanisme pelatihan tersebut dinamakan *backpropagation*/propagasi balik.

Tahap pelatihan ini merupakan langkah bagaimana suatu jaringan saraf itu berlatih, yaitu dengan cara melakukan perubahan pembobot (sambungan antar lapisan yang membentuk jaringan melalui masing-masing unitnya). Sedangkan pemecahan masalah baru akan dilakukan jika proses pelatihan tersebut selesai , fase tersebut adalah fase mapping atau proses pengujian/testing.

**Tabel 2.3** Daftar notasi proses *learning* jaringan saraf tiruan

No	Simbol	Keterangan
1	$x_i$	Unit x ( <i>input</i> ) ke-i pada lapisan <i>input</i>
2	$v_{0j}$	Nilai v (pembobot) pada bias untuk unit $z_j$
3	$v_{ij}$	Nilai v (pembobot) dari unit $x_i$ ke unit $z_j$
4	$z\_in_j$	z (net) <i>input</i> untuk $z_j$
5	$z_j$	Nilai z aktivasi dari unit $z_j$
6	$w_{0k}$	Nilai w (pembobot) pada bias pada <i>output</i> unit $y_k$
7	$w_{jk}$	Nilai w (pembobot) dari $z_j$ ke unit $y_k$
8	$y\_in_k$	<i>Net input</i> untuk $y_k$
9	$y_k$	<i>Output</i> unit ke-k pada lapisan <i>output</i>
10	$t_k$	<i>Output target</i> dari <i>neural network</i>
11	$\delta_j$	Faktor pengaturan nilai pembobot sambungan pada lapisan <i>hidden</i>
12	$\delta_k$	Faktor pengaturan nilai pembobot sambungan pada lapisan <i>output</i>
13	$\alpha$	Konstanta laju pembelajaran ( <i>learning rate</i> ) $0 < \alpha < 1$
14	$\Delta w_{jk}$	Selisih antara $w_{jk}(t)$ dengan $w_{jk}(t + 1)$
15	$\Delta v_{ij}$	Selisih antara $v_{ij}(t)$ dengan $v_{ij}(t + 1)$

Langkah Pelatihan *Backpropagation* terdiri dari dua proses, *feed forward* dan *backpropagation*. Untuk algoritma langkah pelatihan *Backpropagation* sebagai berikut:

- Step 1. Inisialisasi bobot dan masukan dengan menggunakan metode *random gaussian* dengan rata-rata 0 dan deviasi 0,1.
- Step 2. Lakukan langkah 2 hingga 7 untuk masing-masing pasangan data pelatihan *propagasi maju* atau *training data Feedforward*.
- Step 3. Masing-masing unit *input* ( $x_i$ ) untuk  $i = 1, \dots, n$ , menerima sinyal *input*  $x_i$ . Sinyal input disebarkan ke bagian unit lapisan tersembunyi atau *Hidden Layer*.
- Step 4. Tiap unit dilapisan *Hidden Layer* dikalikan dengan faktor pembobot, dijumlahkan dan ditambah dengan biasnya.

$$z\_in_j = v_{0j} + \sum_{i=1}^n x_i v_{ij} \quad (2.9)$$

Kemudian menghitung sesuai dengan fungsi aktivasi yang digunakan:

$$z_j = f(z\_in_j) \quad (2.10)$$

Step 5. Masing-masing unit *output* ( $y_k$ ,  $k=1,2,3\dots m$ ) dikalikan dengan faktor pembobot dan dijumlahkan:

$$y\_in_k = w_{ok} + \sum_{j=1}^p z_j w_{jk} \quad (2.11)$$

Perhitungan kembali fungsi aktifasi

$$y_k = f(y\_in_k) \quad (2.13)$$

*BackPropagasi* dan *error*-nya.

Step 6. Tiap unit *output* ( $y_k$ ) untuk  $k=1,\dots,m$ , menerima pola target sesuai dengan pola *input* saat proses training dan dihitung *error*-nya:

$$\delta_k = (t_k - y_k) f'(y\_in_k) \quad (2.14)$$

Menghitung perbaikan faktor pembobot.  
Kemudian untuk memperbaiki  $w_{jk}$ .

$$\Delta w_{jk} = \alpha \delta_k z_j \quad (2.15)$$

Menghitung perbaikan bias koreksi bias:

$$\Delta w_{0k} = \alpha \delta_k \quad (2.16)$$

dan menggunakan nilainya pada semua unit lapisan sebelumnya.

Step 7. Masing-masing pembobot yang menghubungkan unit-unit lapisan *output* dengan unit-unit pada lapisan tersembunyi ( $z_{j,j=1\dots p}$ ) dikalikan delta dan dijumlahkan sebagai *input* ke unit-unit lapisan berikutnya.



$$\delta_{in_j} = \sum_{k=1}^m \delta_k w_{jk} \quad (2.17)$$

Selanjutnya dikalikan dengan turunan dari fungsi aktivasinya untuk menghitung *error*nya.

$$\delta_j = \delta_{in_j} f'(y_{in_j}) \quad (2.18)$$

Kemudian menghitung perbaikan pembobot digunakan untuk memperbaiki  $v_{ij}$ .

$$\Delta v_{ij} = \alpha \delta_j x_i \quad (2.19)$$

Kemudian menghitung perbaikan bias untuk Memperbaiki  $v_{0j}$

$$\Delta v_{0j} = \alpha \delta_j \quad (2.20)$$

Masing-masing *output* unit ( $y_k$ ), untuk  $k=1,...,m$  diperbaiki bias dan pembobotnya ( $j=0,...,p$ )

$$w_{jk}(baru) = w_{jk}(lama) + \Delta w_{jk} \quad (2.21)$$

Tiap unit tersembunyi ( $z_j$ ,  $j: 1,...,p$ ) diperbaiki bias dan pembobotnya untuk  $j=0,...,n$ .

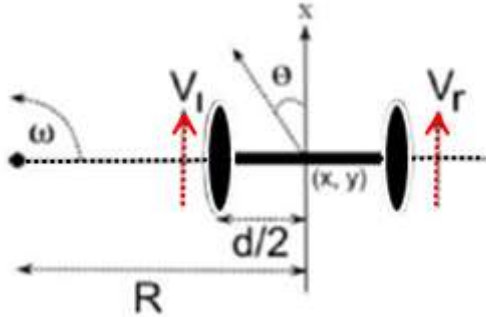
$$v_{ij}(baru) = v_{ij}(lama) + \Delta v_{ij} \quad (2.22)$$

Step 8. Uji kondisi pemberhentian (akhir iterasi).

Step 9. Ulangi langkah 1 hingga 8 sampai kondisi akhir iterasi dipenuhi

## 2.7. Kinematika *Mobile Robot Qbot*

Kinematika pada *mobile robot* ini membahas gerak robot dan sistem robot beroperasi dalam pergerakan tanpa mempersoalkan gaya penyebab gerakan. Kinematika pada *mobile robot Qbot* yaitu *forward kinematics* dan *invers kinematics* [4].



**Gambar 2.25** *Differential Kinematic of Mobile Robot Qbot [4]*

Motor pada kedua roda *mobile robot* dapat dikendalikan dengan cara mengatur kecepatan pada kedua roda. Kecepatan rotasi dilambangkan  $\omega$  sekitar ICC (*Instantaneous Center of Curvature*) harus sama pada kedua roda. Maka, persamaan berikut ini menyusun hubungan antara parameter gerak dari *differential drive mobile robot*.

$$\omega \left( R + \frac{d}{2} \right) = V_{right} \quad (2.23)$$

$$\omega \left( R - \frac{d}{2} \right) = V_{left} \quad (2.24)$$

Keterangan:

1.  $d$  adalah jarak antara titik pusat dari kedua roda
2.  $R$  adalah jarak dari ICC ke titik tengah antar roda
3.  $V_{right}$  dan  $V_{left}$  adalah kecepatan roda kanan dan kiri

Kemudian persamaan 2.23 dan 2.24 digunakan untuk setiap kasus waktu untuk  $R$  dan  $\omega$  sebagai berikut. Mencari  $R$  menggunakan persamaan sebagai berikut.

$$V_{right} = \omega \left( R + \frac{d}{2} \right) \rightarrow \omega = \frac{V_{right}}{\left( R + \frac{d}{2} \right)} \quad (2.25)$$

$$V_{left} = \omega \left( R - \frac{d}{2} \right) \rightarrow \omega = \frac{V_{left}}{\left( R - \frac{d}{2} \right)} \quad (2.26)$$

$$\frac{V_{right}}{\left(R + \frac{d}{2}\right)} = \frac{V_{left}}{\left(R + \frac{d}{2}\right)} \quad (2.27)$$

$$V_{right} \left(R + \frac{d}{2}\right) = V_{left} \left(R - \frac{d}{2}\right) \quad (2.28)$$

$$V_{right} R - \frac{V_{right} d}{2} = V_{left} R - \frac{V_{left} d}{2} \quad (2.29)$$

$$V_{right} R - V_{left} R = \frac{V_{right} d}{2} + \frac{V_{left} d}{2} \quad (2.30)$$

$$R (V_{right} - V_{left}) = \frac{d}{2} (V_{right} + V_{left}) \quad (2.31)$$

$$R = \frac{d(V_{right} + V_{left})}{2(V_{right} - V_{left})} \quad (2.32)$$

Selanjutnya mencari Persamaan  $\omega$  untuk mencari persamaan 2.33,

$$V_{right} = \omega \left(R + \frac{d}{2}\right) \rightarrow \omega R = V_{right} - \omega \frac{d}{2}$$

$$V_{left} = \omega \left(R - \frac{d}{2}\right) \rightarrow \omega R = V_{left} + \omega \frac{d}{2}$$

$$V_{right} - \omega \frac{d}{2} = V_{left} + \omega \frac{d}{2}$$

$$2 \omega \frac{d}{2} = V_{right} - V_{left}$$

$$\omega d = V_{right} - V_{left}$$

$$\omega = \frac{V_{right} - V_{left}}{d} \quad (2.33)$$

Dari Persamaan 2.25 dan Persamaan 2.26 dapat diketahui hubungan antara  $R$  dan  $\omega$  untuk mencari kecepatan linier ( $V$ ).

$$V = \omega x R$$

$$V = \frac{V_{right} - V_{left}}{d} \times \frac{d(V_{right} + V_{left})}{2(V_{right} - V_{left})}$$

$$V = \frac{V_{right} + V_{left}}{2} \quad (2.34)$$

Pada Gambar 2.26 atas bahwa robot berada di sebuah posisi (x,y) dan menuju ke arah posisi target(x,y) yang membuat sudut  $\theta$  dengan X aksis.

Dengan mengetahui kecepatan  $V_{right}$ ,  $V_{left}$  dan menggunakan Persamaan 2.25 dan Persamaan 2.26 lokasi ICC ( $ICC_x$  ,  $ICC_y$ ) dapat ditentukan pada Persamaan 2.33 dan Persamaan 2.34.

$$ICC_x = x - R \sin \theta = x - \frac{d(V_{right} + V_{left})}{2(V_{right} - V_{left})} \sin \theta \quad (2.35)$$

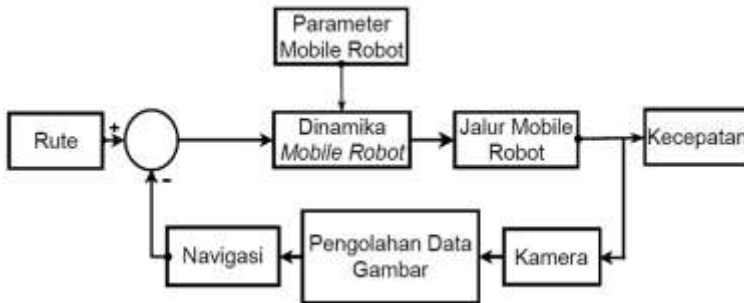
$$ICC_y = y + R \cos \theta = y + \frac{d(V_{right} + V_{left})}{2(V_{right} - V_{left})} \cos \theta \quad (2.34)$$

## BAB III PERANCANGAN SISTEM

Pembahasan pada bab ini mengenai proses perancangan sistem, yaitu merancang simulasi dan implementasi perencanaan jalur *mobile robot* Qbot. Perancangan algoritma untuk perencanaan jalur dengan menggunakan metode Jaringan Saraf Tiruan sebagai pengenalan pola berbasis data gambar.

### 3.1. Perancangan Navigasi Pada *Mobile Robot Qbot*

Diagram blok pada Gambar 3.1 berikut merupakan sistem keseluruhan dari pemrosesan penerapan navigasi pada mobile robot.



**Gambar 3.1** Diagram Blok Sistem

Keterangan Blok Diagram Sebagai berikut.

1. *Rute*
2. Parameter *mobile robot* adalah parameter – parameter dari *mobile robot*
3. *Dinamika mobile robot* merupakan step dalam melakukan pemodelan sistem berupa pendekatan matematis dan dasar – dasar dinamika dari *mobile robot*.
4. *Jalur mobile robot*
5. Kamera sebagai sensor yang akan mengambil data gambar dari jalur dan benda.
6. Pengolahan data gambar merupakan tempat terjadi proses data gambar untuk dijadikan kondisi terhadap *mobile*

robot *Qbot*. Kondisi berupa pertigaan, perempatan, belok dan saat jalan lurus.

7. Navigasi Petunjuk arah mobile robot melaju. Navigasi disusun sederhana agar robot mampu mengenali maksud dari arah robot melaju.
8. Perhitungan Sudut dan kecepatan roda kanan dan kiri

### 3.2 Desain *Kinematics*

Berikut merupakan desain untuk pergerakan pada mobile robot.

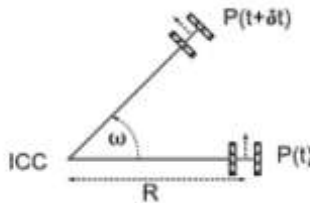
#### 3.2.1 Desain *Forward Kinematics*

*Forward kinematics* digunakan untuk permasalahan kecepatan pada roda dan konfigurasi posisi awal robot  $(x, y, \theta)_{t=0}$  dapat menentukan posisi robot  $(x, y, \theta)$ .

Kecepatan yang tidak bervariasi pada  $V_{right}$  dan  $V_{left}$ , lokasi ICC ( $ICC_x, ICC_y$ ) akan menjadi posisi tetap dikenal sebagai *Pose Relative Robot* pada Posisi Awal Robot. Karena itu, pada waktu  $t+\delta t$  pose robot akan menjadi:

$$\begin{bmatrix} x' \\ y' \\ \theta' \end{bmatrix} = \begin{bmatrix} \cos(\omega \times \delta t) & -\sin(\omega \times \delta t) & 0 \\ \sin(\omega \times \delta t) & \cos(\omega \times \delta t) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x - ICC_x \\ y - ICC_y \\ \theta \end{bmatrix} + \begin{bmatrix} ICC_x \\ ICC_y \\ \omega \times \delta t \end{bmatrix} \quad (3.1)$$

Dimana  $(x, y, \theta)$  dan  $(x', y', \theta')$  adalah posisi robot pada waktu  $t$  dan  $t+\delta t$ , berturut-turut. Persamaan ini menggambarkan gerak berputar robot sekitar ICC nya dengan sebuah radius dari lengkungan  $R$  dan sebuah kecepatan sudut  $\omega$ , seperti pada Gambar 3.2.



**Gambar 3.2** *Forward Kinematics Relative to ICC*

Persamaan gerak umum robot yang mampu bergerak dalam arah  $\theta(t)$  tertentu pada kecepatan  $V(t)$  yang diberikan, dijelaskan sebagai berikut,

$$\begin{aligned}x(t) &= \int_0^t V(t) \cos[\theta(t)] dt \\y(t) &= \int_0^t V(t) \sin[\theta(t)] dt \\\theta(t) &= \int_0^t \omega(t) dt\end{aligned}\tag{3.2}$$

Untuk sebuah robot *differential drive* seperti Qbot, Persamaan 3.2 menjadi,

$$\begin{aligned}x(t) &= \frac{1}{2} \int_0^t [V_r(t) + V_l(t)] \cos[\theta(t)] dt \\y(t) &= \frac{1}{2} \int_0^t V(t) \sin[\theta(t)] dt \\\theta(t) &= \frac{1}{d} \int_0^t \omega(t) dt\end{aligned}\tag{3.3}$$

Persamaan 3.2 dapat disederhanakan untuk menentukan *pose* robot saat  $V$  sebagai berikut,

$$\begin{bmatrix} x' \\ y' \\ \theta' \end{bmatrix} = \begin{bmatrix} x + V \cos(\theta) \delta t \\ y + V \sin(\theta) \delta t \\ \theta + \omega \delta t \end{bmatrix}\tag{3.4}$$

Demikian pula pada Persamaan 3.3,

$$\begin{bmatrix} x' \\ y' \\ \theta' \end{bmatrix} = \begin{bmatrix} x + \frac{1}{2} [V_r + V_l] \cos(\theta) \delta t \\ y + \frac{1}{2} [V_r + V_l] \sin(\theta) \delta t \\ \theta + \frac{1}{d} [V_r - V_l] \delta t \end{bmatrix}\tag{3.5}$$

Dengan demikian, untuk kasus khusus dari  $V_{left} = V_{right} = V$  dan  $V_{left} = V_{right} = V$ , Persamaan 3.3 dan Persamaan 3.4 menjadi Persamaan 3.5 dan Persamaan 3.6.

$$\begin{bmatrix} x' \\ y' \\ \theta' \end{bmatrix} = \begin{bmatrix} x + V \cos(\theta) \delta t \\ y + V \sin(\theta) \delta t \\ \theta \end{bmatrix}\tag{3.6}$$

$$\begin{bmatrix} x' \\ y' \\ \theta' \end{bmatrix} = \begin{bmatrix} x \\ y \\ \theta + \frac{2V\delta t}{d} \end{bmatrix} \quad (3.7)$$

### 3.2.2. Desain Inverse Kinematics

Berikut ini penjelasan Permasalahan *inverse kinematics* tentang bagaimana kecepatan yang diberikan pada roda dan konfigurasi posisi awal robot  $(x, y, \theta)_{t=0}$  dapat menentukan posisi robot  $(x, y, \theta)$ . Diasumsikan  $V_{right}(t) = V_{right}$ ,  $V_{left}(t) = V_{left}$  dan  $(x, y, \theta)_{t=0} = (0,0,0)$ . Seperti Persamaan 3.8.

$$\begin{aligned} x(t) &= \frac{d(V_{right}+V_{left})}{2(V_{right}-V_{left})} \sin \left[ \frac{t}{d} (V_{right} - V_{left}) \right] \\ y(t) &= -\frac{d(V_{right}+V_{left})}{2(V_{right}-V_{left})} \cos \left[ \frac{t}{d} (V_{right} - V_{left}) \right] + \frac{d(V_{right}+V_{left})}{2(V_{right}-V_{left})} \\ \theta(t) &= \frac{d(V_{right}+V_{left})}{2(V_{right}-V_{left})} \end{aligned} \quad (3.8)$$

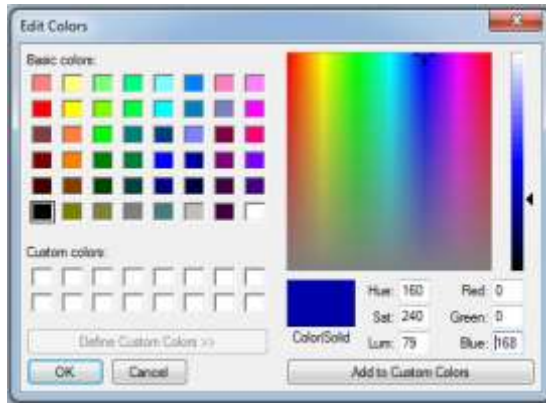
Diberikan waktu tujuan (t) dan posisi target (x,y), pada Persamaan 3.7 menyederhanakan  $V_{right}$  dan  $V_{left}$ , tetapi tidak menyediakan kontrol sendiri pada  $\theta$ .

Persamaan 3.5 dan Persamaan 3.6 melakukan strategi sederhana untuk mendorong robot ke posisi tujuan untuk mendorong robot untuk beberapa tujuan menimbulkan  $(x, y, \theta)$ , robot bisa berputar sampai ke tempat tujuan pada(x, y), kemudian berjalan foward sampai titik (x, y), dan kemudian memutar di tempat sampai orientasi tujuan yang diperlukan  $\theta$  terpenuhi.

### 3.3 Pengaturan Warna pada Obyek

Penentuan warna obyek yang akan dianalisa oleh robot pada gambar bisa diketahui parameternya menggunakan aplikasi *edit color* milik *Paint Tool Microsoft*. Pada perancangan ini kita menggunakan warna biru sebagai obyek yang akan diidentifikasi.





**Gambar 3.3** Pengaturan Warna Pada *Color Path*

Pada *Cursor* geser di atas nilai *red*, *green* dan *Blue* digeser mencari warna yang sesuai dengan obyek. Tujuannya adalah menyeleksi nilai warna pada obyek yang menjadi elemen dasar pada gambar. Elemen ini yang akan dilakukan *image processing*.

Setelah pengaturan geser *cursor item* maka akan tampil nilai RGB. Nilai RGB diambil untuk diterapkan pada program sebagai pembatasan warna.

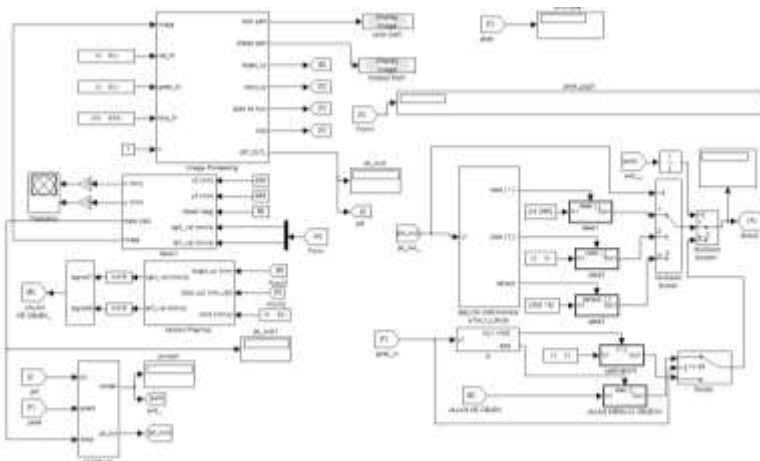
### 3.4. Prinsip Kerja Sistem

Pengambilan data gambar pada obyek yang berwarna biru seperti telah dijelaskan pada sub-bab 3.3 yang hasilnya diproses ke bentuk hitam putih. Labelisasi dibutuhkan karena robot menganalisis hanya gambar yang dinyatakan obyek. Setelah obyek dilabelisasi maka sistem diberi kemampuan untuk mengenali bentuk obyek. Pengenalan pola adalah cara yang baik untuk mengidentifikasi obyek, dalam khusus ini pengenalan pola menggunakan metode jaringan saraf tiruan untuk membedakan setiap bentuk obyek. Bentuk obyek menjadi navigasi robot sehingga terbentuk sistem navigasi yang menentukan arah robot beroperasi. Hasil dari navigasi akan diproses robot disebut *motion planning* robot. Perancangan sistem ini dibentuk ke dalam bentuk *flowchart* seperti pada gambar 3.4.

Perancangan simulasi dalam flowchart seperti pada Gambar 3.4.



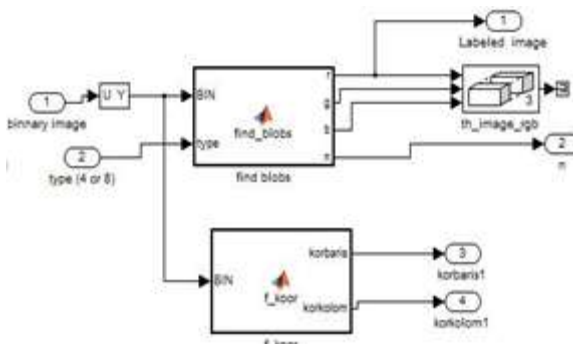
**Gambar 3.4** Perancangan Pengambilan *Image* Hingga *Motion Planing*



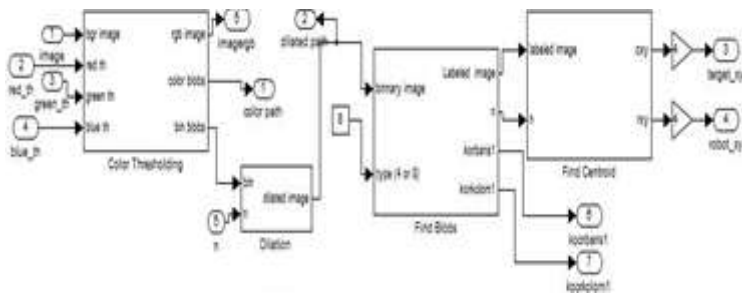
**Gambar 3.5** Rancangan Simulink Secara Keseluruhan

Keterangan:

1. Blok *image* processing merupakan tempat pemrosesan data *image*
2. Blok *Qbot* merupakan pemrosesan rencana gerakan atau motion planner
3. Blok *motion planner* rancangan untuk memberikan keputusan jalur yang harus dilalui robot menuju titik target yang diinginkan.



**Gambar 3.6** Rancangan Simulink Untuk Koordinat Target

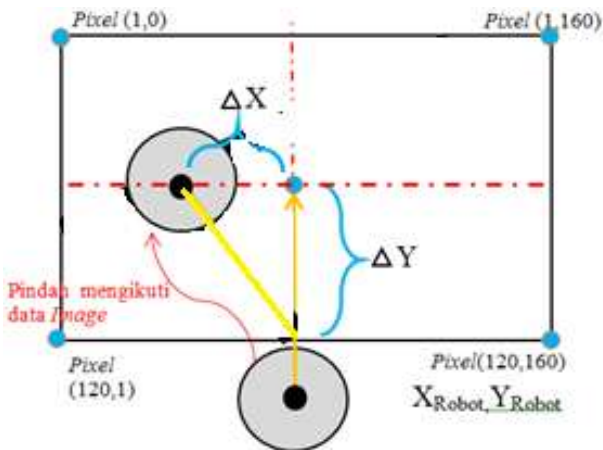


**Gambar 3.7** Rancangan Simulink Untuk Color Tresh hold, Find Bolb hingga robot Target

### 3.5. Rancangan Navigasi *Mobile Robot*

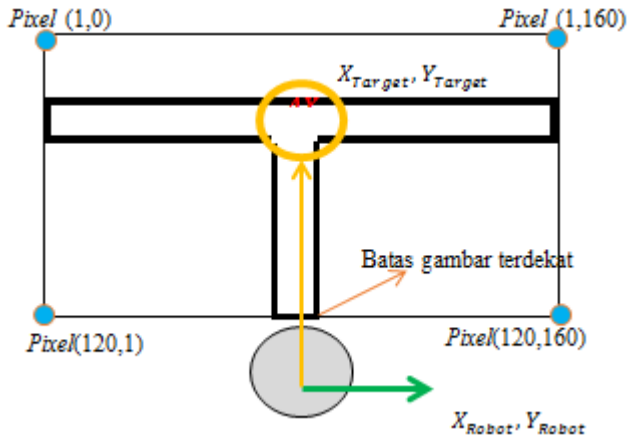
Rancangan navigasi yang digunakan adalah menggunakan metode koreksi koordinat baris dan kolom untuk mendeteksi adanya pertigaan atau tidak dan kemudian metode ini mengatur sudut kendali.

Kamera mobile robot bernilai *pixel* 160 x 120 obyek berupa jalur pertigaan, jalur lurus atau jalur perempatan. Rancangan sistem pendeteksian pada pertigaan jalur seperti pada Gambar 3.8.



**Gambar 3.8** Perpindahan Posisi Robot Ke Posisi Target

*Mobile robot Qbot* menggunakan analisa piksel dalam bentuk matrik seperti pada Gambar 3.9.



**Gambar 3.9** Posisi Robot Sekarang dan Posisi Target

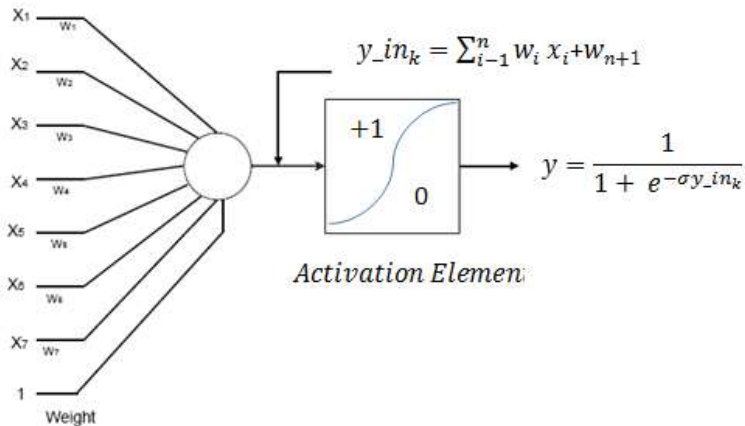
Keterangan:

1.  $X_{Robot}$  adalah robot berada pada posisi koordinat X robot saat sekarang
2.  $Y_{Robot}$  adalah robot berada pada posisi koordinat Y robot saat sekarang
3.  $X_{Target}$  adalah koordinat X yang akan dituju oleh robot pada perempatan

$Y_{Robo}$  adalah koordinat Y yang akan dituju oleh robot pada perempatan

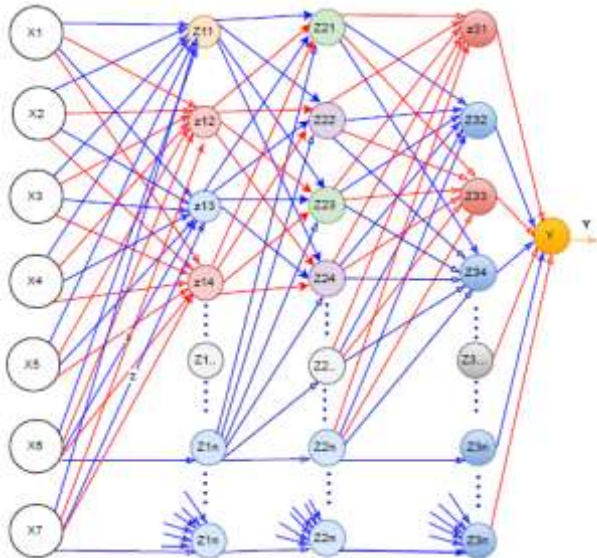
### 3.6. Struktur Jaringan Saraf Tiruan [2]

Sistem navigasi menjadi petunjuk arah pada *mobile robot* secara otomatis melewati lintasan jalur tanpa keluar dari yang ditentukan. Pengujian – pengujian yang dilakukan adalah pengujian *image processing* berupa gambar hasil dari metode jaringan saraf tiruan pada setiap kondisi yang sulit. Sehingga dirancang metode jaringan saraf tiruan agar robot dapat memutuskan arah yang benar.



**Gambar 3.9** Struktur Jaringan Saraf Tiruan dalam satu node

Untuk perancangan jaringan saraf tiruan secara keseluruhan adalah seperti Gambar 3.10 berikut. [3]



**Gambar 3.10** Rancangan Jaringan Saraf Tiruan Secara Keseluruhan

### 3.6.1. Perhitungan *Feedforward*

Perhitungan *Feedforward* adalah sebagai berikut.

Langkah 1. Masing-masing unit *input* ( $x_i$ ) untuk  $i = 1, \dots, n$ , menerima sinyal *input*  $x_i$ . Sinyal tersebut disebarkan ke tiap bagian unit lapisan *hidden layer*.

Langkah 2. Tiap unit *hidden layer* dikalikan dengan faktor pembobot, dijumlahkan dan ditambah dengan biasnya:

$$z\_in_j = v_{oj} + \sum_{i=1}^n x_i v_{ij} \quad (3.9)$$

Kemudian menghitung fungsi aktivasi.

$$z_j = f(z\_in_j) \quad (3.10)$$

Langkah 3. Tiap unit *output* ( $y_k$ ,  $k=1,2,3,\dots,m$ ) dikalikan dengan faktor pembobot dan dijumlahkan:

$$y\_in_k = w_{ok} + \sum_{j=1}^p z_j w_{jk} \quad (3.11)$$

Perhitungan fungsi aktivasi  $y_k$ .

$$y_k = f(y\_in_k)$$

### 3.6.2. Perhitungan *Backward (Backpropagation)*

Setelah melakukan perhitungan *Feedforward*, maka akan muncul data yang berupa besar sudut *joint* manipulator robot, dimana besar sudut *joint* yang dihasilkan *inverse kinematics neural network* belum tentu sesuai dengan target tujuan, oleh karena itu dilakukan perhitungan *backward* atau *backpropagation* dimana perhitungan ini mencari error perhitungan yang dilakukan oleh *Feedforward* dan melakukan revisi bobot pada *inverse kinematics neural network* dan melakukan perhitungan *Feedforward* lagi sampai menemukan error terkecil atau kesesuaian besar sudut *joint* dengan target yang diinginkan.

*BackPropagasi dan errornya*

Langkah 4. Masing-masing unit *output* ( $y_k$ ,  $k=1,...,m$ ) menerima pola target sesuai dengan pola *input* saat pelatihan/training dan dihitung *error*nya:

$$\delta_k = (t_k - y_k)f'(y_{in_k}) \quad (3.12)$$

Menghitung perbaikan faktor pembobot (kemudian untuk memperbaiki  $w_{jk}$ ):

$$\Delta w_{jk} = \alpha \delta_k z_j \quad (3.13)$$

Menghitung perbaikan bias koreksi bias:

$$\Delta w_{0k} = \alpha \delta_k \quad (3.14)$$

dan menggunakan nilainya pada semua unit lapisan sebelumnya.

Langkah 5. Masing-masing pembobot yang menghubungkan unit-unit lapisan *output* dengan unit-unit pada lapisan tersembunyi ( $z_j, j=1..,p$ ) dikalikan delta dan dijumlahkan sebagai *input* ke unit-unit lapisan berikutnya.

$$\delta_{in_j} = \sum_{k=1}^m \delta_k w_{jk} \quad (3.15)$$

Selanjutnya dikalikan dengan turunan dari fungsi aktivasinya untuk menghitung *error*nya.

$$\delta_j = \delta_{in_j} f'(y_{in_j}) \quad (3.16)$$

Kemudian menghitung perbaikan pembobot (digunakan untuk memperbaiki  $v_{ij}$ ).

$$\Delta v_{ij} = \alpha \delta_j x_i \quad (3.17)$$



Kemudian perbaiki bias (untuk memperbaiki  $v_{0j}$ )

$$\Delta v_{0j} = \alpha \delta_j \quad (3.18)$$

Langkah 6. Tiap *output* unit  $y_k$ , untuk  $k=1, \dots, m$  diperbaiki bias dan pembobotnya ( $j=0, \dots, p$ )

$$w_{jk}(\text{baru}) = w_{jk}(\text{lama}) + \Delta w_{jk} \quad (3.19)$$

Tiap unit tersembunyi  $z_j$ , untuk  $j: 1, \dots, p$  diperbaiki bias dan pembobotnya ( $j=0, \dots, n$ ).

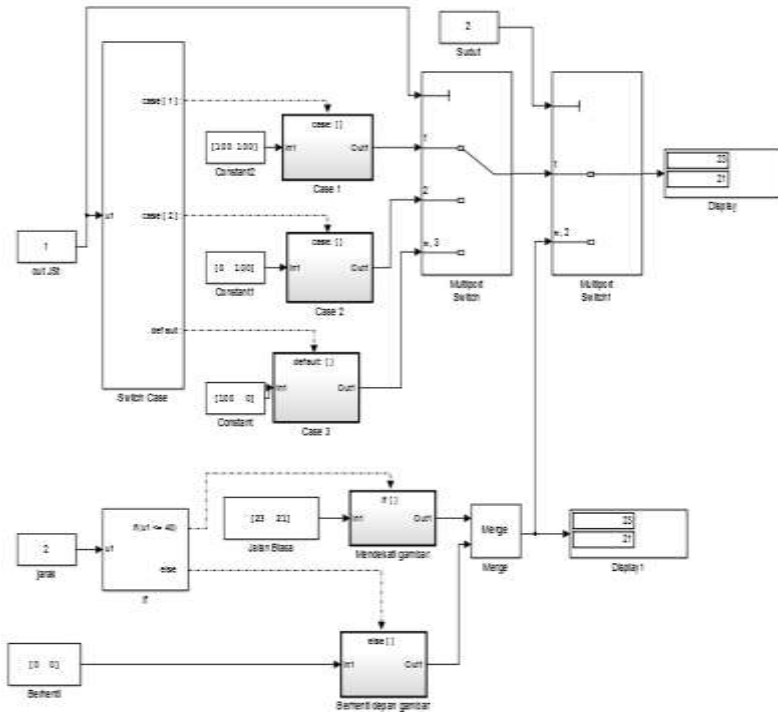
$$v_{ij}(\text{baru}) = v_{ij}(\text{lama}) + \Delta v_{ij} \quad (3.20)$$

Proses pemisahan Background dengan obyek yang dijadikan input untuk jaringan saraf tiruan adalah sebagai berikut.

Jaringan saraf tiruan digunakan untuk melihat rambu-rambu apakah belok kanan, belok kiri atau berhenti dan kapan harus jalan lurus. Seperti fungsi navigasi yaitu petunjuk arah yang benar pada sebuah jalan raya di kehidupan nyata. Robot mendekati setiap simbol navigasi sebelum memutuskan keputusan arah yang akan dituju oleh *mobile robot*. Pada rancangan rangkaian dibawah adalah bentuk struktur dari sistem navigasi. Pada Gambar 3.11 dirancang pengelompokan nilai untuk menjadi *port* yang akan menentukan kecepatan robot saat berbelok kiri kekanan maupun lurus sesuai navigasi yang telah direncanakan.

Algoritma program dijadikan dalam bentuk *per-step* dalam bahasa manusia sebagai berikut adalah sebagai berikut. [4]

- Step 1.* Tekan tombol Start, jika jarak lebih dari 40 iterasi atau kecil dari 15 cm ke obyek yang dideteksi maka jalan biasa.
- Step 2.* Jarak piksel ke benda kurang dari 40 iterasi atau kurang dari 10 cm maka robot berhenti.



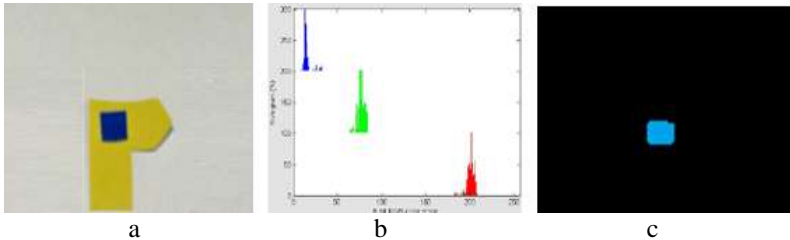
**Gambar 3.11** Rancangan Navigasi

Ketika robot berhenti maka robot melakukan pengecekan navigasi apakah belok kanan, belok kiri atau jalan lurus. *Step* pengecekan adalah sebagai berikut.

- Step 1.* Jika ditemukan *navigasi* tanda belok kanan maka robot belok kanan hingga ditemukan target yang akan dituju selanjutnya.
- Step 2.* Jika ditemukan *navigasi* tanda belok kiri maka robot belok kanan hingga ditemukan target yang akan dituju selanjutnya.
- Step 3.* Jika ditemukan navigasi tanda jalan lurus maka navigasi dilewati dengan jalan lurus hingga ditemukan target koordinat selanjutnya.
- Step 4.* Dan seterusnya sistem akan terus berjalan.

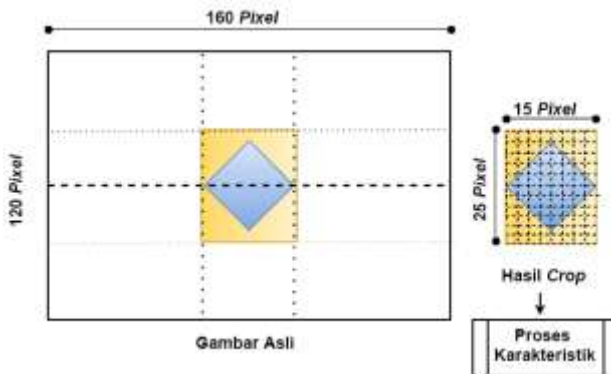
### 3.7 Pemisah Obyek dengan *Background* Gambar

Pemisahan *background* dengan gambar tujuannya untuk hanya menganalisa bentuk gambar yang kita inginkan. Pada Gambar 3.12 yang ditampilkan adalah warna biru saja dan bukan warna biru di hitamkan. [5]



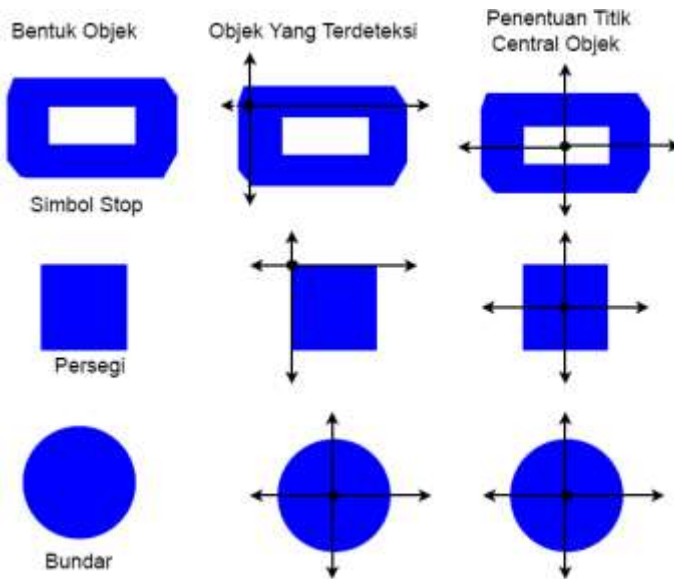
**Gambar 3.12** (a) Gambar Asli (b) Pengambilan Biru Pada Warna RGB (c) Menampilkan Hanya Warna Biru

Proses selanjutnya adalah *Crop Image*. Seperti pada gambar 3.13 bahwa robot hanya memproses gambar yang berwarna biru dan selain itu tidak.



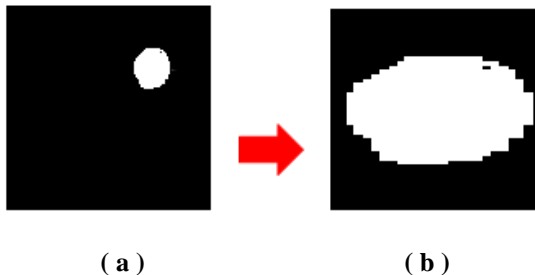
**Gambar 3.13** Proses *Image Crop*

Sebelum memisahkan hasil crop dari gambar asli maka dibutuhkan penentuan titik tengah dari gambar yang akan di-*crop* seperti Gambar 4.15.



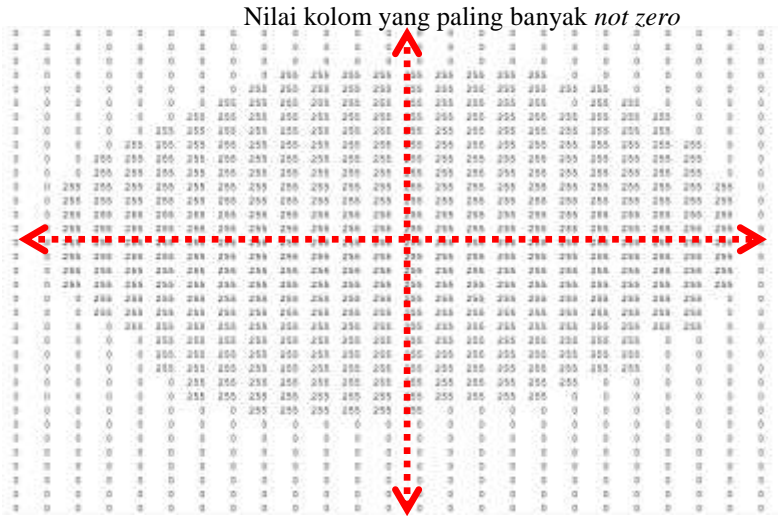
**Gambar 3.14** Proses Penentuan Titik *Central Crop* Obyek

Proses Peng-ambangan [6] merupakan pengaturan batas-batas nilai ambang yang digunakan untuk memperoleh citra biner yang memiliki warna putih dan hitam seperti Gambar 3.15.



**Gambar 3.15** (a) adalah Bentuk Bundar Yang akan Di-*Crop* dan (b) Hasil *Crop*

Program *Crop* dilampirkan pada Lampiran A.1 yang menghasilkan keluaran yang bernilai seperti Gambar 3.16.



**Gambar 3.16** Analisa Untuk Crop gambar segitiga ditampilkan seperti pada gambar di atas

Program *Crop* dilampirkan pada Lampiran A.1 Untuk menganalisis karakteristik gambar yang telah di-*crop* maka dibutuhkan momen *invariant* untuk menyatakan obyek dengan memperhitungkan area obyek. Momen ini menggunakan momen spasial dan momen pusat kedua momen ini nantinya digunakan untuk menangani translasi, penekalaan, dan rotasi gambar yang dinamakan momenhu. Penciptanya teori ini yaitu Hu (Theodoridis dan Koutroubus,2006),

Dalam pengolahan citra momen spasial digunakan untuk menganalisis orde dua dimensi untuk menyatakan jumlah baris dan kolom, dengan orde (m,n) yang didefinisikan sebagai berikut :

$$M_{ij} = \sum_{x=1}^M \sum_{y=1}^N x^i y^j I(x,y) \quad (3.21)$$

Keterangan,

1. i dan j menyatakan orde momen untuk  $i = 0,1,2,\dots$ , dan  $j = 0,1,2,3,\dots$
2. M menyatakan jumlah kolom pada citra

3. N menyatakan jumlah baris pada citra.
4. X adalah ordinat piksel
5. Y adalah absis piksel
6. I(x,y) menyatakan intensitas piksel pada posisi (x,y)

Rancangan program untuk momen spasial adalah seperti *script* berikut.

```
function [hasil] = momen_spasial(F, p, q)
[m, n] = size(F);
momenPQ = 0;
for y=1 : m
    for x=1 : n
        if F(y,x) ~= 0
            momenPQ = momenPQ + x^p * y^q;
        end
    end
end
hasil = momenPQ;
```

Adapun momen pusat adalah momen spasial yang dihitung relatif terhadap pusat massa. Jika pusat massa adalah  $(\bar{y}, \bar{x})$ , momen pusat ditulis sebagai berikut :

$$\mu_{ij} = \sum_{x=1}^M \sum_{y=1}^N (x - \bar{x})^i (y - \bar{y})^j I(x, y) \quad (3.22)$$

Untuk rancangan program momen pusat adalah seperti *script* berikut.

```
function [hasil] = momen_pusat(F, p, q)
% MOMEN_PUSAT Menghitung moment pusat berorde p, q
[m, n] = size(F);
m00 = momen_spasial(F, 0, 0);
xc = momen_spasial(F, 1, 0) / m00;
yc = momen_spasial(F, 0, 1) / m00;
mpq = 0;
for y=1 : m
    for x=1 : n
        if F(y,x) ~= 0
```

```

        mpq = mpq + (x-xc)^p * (y-yc)^q;
    end
end
end
hasil = mpq;

```

Momen di atas bersifat tidak terpengaruh terhadap transisi. Dalam hal ini,  $\bar{x}$  dan  $\bar{y}$  diperoleh melalui :

$$\bar{x} = \frac{M_{10}}{M_{00}}, \bar{y} = \frac{M_{01}}{M_{00}} \quad (3.23)$$

Agar momen pusat bersifat bebas terhadap translasi, penyekalaan, dan rotasi maka momen perlu di dinormalisasi. Momen pusat ternormalisasi berupa:

$$\eta_{ij} = \frac{\mu_{pq}}{\mu_{00}^{\frac{p+q}{2}}}, \gamma = \frac{i+j+2}{2} \quad (3.24)$$

Rancangan program momen pusat adalah seperti script berikut.

```

function [hasil] = normomen(F, p, q)
% NORMOMEN Menghitung moment pusat ternormalisasi.
%   Masukan: F = Citra biner.
%           p dan q = orde momen.
F = double(F);
m00 = momen_spasial(F, 0, 0);
normalisasi = m00 ^ ((p+q+2)/2.0);
hasil = momen_pusat(F, p, q) / normalisasi;
function [hasil] = momen_pusat(F, p, q)
% Menghitung moment pusat berorde p, q
[m, n] = size(F);
m00 = momen_spasial(F, 0, 0);
xc = momen_spasial(F, 1, 0) / m00;
yc = momen_spasial(F, 0, 1) / m00;
mpq = 0;
for y=1 : m
    for x=1 : n
        if F(y,x) ~= 0
            mpq = mpq + (x-xc)^p * (y-yc)^q;
        end
    end
end
end

```

```

    end
  end
end
hasil = mpq;

```

Momen yang dihasilkan dapat digunakan untuk menangani translasi, penyekalaan, dan rotasi pada gambar. Dikenal sebagai momenhu. Momen ini memiliki 7 momen invariant seperti berikut.

$$\theta_1 = \eta_{20} + \eta_{02} \quad (3.25)$$

$$\theta_2 = (\eta_{20} + \eta_{02})^2 + 2\eta_{02}^2 \quad (3.26)$$

$$\theta_3 = (\eta_{30} + 3\eta_{12})^2 + (\eta_{03} - 3\eta_{21})^2 \quad (3.27)$$

$$\theta_4 = (\eta_{30} + 3\eta_{12})^2 + (\eta_{03} - 3\eta_{21})^2 \quad (3.28)$$

$$\theta_5 = (\eta_{30} + 3\eta_{12})(\eta_{03} - 3\eta_{21}) [(\eta_{30} + 3\eta_{12})^2 - 3(\eta_{21} - 3\eta_{03})^2] \\ + (\eta_{03} - 3\eta_{12})(\eta_{03} + 3\eta_{21}) [(\eta_{30} + 3\eta_{12})^2 - 3(\eta_{12} - \eta_{30})^2] \quad (3.29)$$

$$\theta_6 = (\eta_{20} + 3\eta_{02})[(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2] \\ + 4\eta_{11}(\eta_{30} + \eta_{12})(\eta_{03} - 3\eta_{21}) \quad (3.30)$$

$$\theta_7 = (3\eta_{21} - \eta_{03})(\eta_{30} + 3\eta_{12})[(\eta_{30} + 3\eta_{12})^2 - 3(\eta_{21} + \eta_{03})^2] \\ (\eta_{30} + \eta_{12})(\eta_{21} + 3\eta_{03})[(\eta_{03} - 3\eta_{21})^2 - 3(\eta_{30} + 3\eta_{12})^2] \quad (3.31)$$

Program momenhu adaah seperti *script* berikut.

```

function [Momen] = momenhu(F)
% MOMENHU Menghitung momen HU.
%   Masukan: F = citra berskala keabuan
%   Keluaran: Momen = 7 momen Hu

norm_20 = normmomen(F, 2, 0);
norm_02 = normmomen(F, 0, 2);
norm_11 = normmomen(F, 1, 1);
norm_30 = normmomen(F, 3, 0);
norm_12 = normmomen(F, 1, 2);
norm_21 = normmomen(F, 2, 1);
norm_03 = normmomen(F, 0, 3);

Momen.m1 = norm_20 + norm_02;

```



$$\text{Momen.m2} = (\text{norm\_20} - \text{norm\_02})^2 + 4 * \text{norm\_11}^2;$$

$$\text{Momen.m3} = (\text{norm\_30} + 3 * \text{norm\_12})^2 + (3 * \text{norm\_21} - \text{norm\_03})^2;$$

$$\text{Momen.m4} = (\text{norm\_30} + \text{norm\_12})^2 + (\text{norm\_21} + \text{norm\_03})^2;$$

$$\begin{aligned} \text{Momen.m5} = & (\text{norm\_30} - 3 * \text{norm\_12}) * (\text{norm\_30} + \text{norm\_12}) * ... \\ & ((\text{norm\_30} + \text{norm\_12})^2 - 3 * (\text{norm\_21} + \text{norm\_03})^2) + ... \\ & (3 * \text{norm\_21} - \text{norm\_03}) * (\text{norm\_21} + \text{norm\_03}) * ... \\ & (3 * (\text{norm\_30} + \text{norm\_12})^2 - (\text{norm\_21} + \text{norm\_03})^2); \end{aligned}$$

$$\begin{aligned} \text{Momen.m6} = & (\text{norm\_20} - \text{norm\_02}) * ((\text{norm\_30} + \text{norm\_12})^2 - ... \\ & (\text{norm\_21} + \text{norm\_03})^2) + ... \\ & 4 * \text{norm\_11} * (\text{norm\_30} + \text{norm\_12}) * ... \\ & (\text{norm\_21} + \text{norm\_03}); \end{aligned}$$

$$\begin{aligned} \text{Momen.m7} = & (3 * \text{norm\_21} + \text{norm\_30}) * (\text{norm\_30} + \text{norm\_12}) * ... \\ & ((\text{norm\_30} + \text{norm\_12})^2 - 3 * (\text{norm\_21} + \text{norm\_03})^2) + ... \\ & (\text{norm\_30} - 3 * \text{norm\_12}) * (\text{norm\_21} + \text{norm\_03}) * ... \\ & (3 * (\text{norm\_30} + \text{norm\_12})^2 - (\text{norm\_21} + \text{norm\_03})^2); \end{aligned}$$

*Halaman ini sengaja dikosongkan*

## BAB IV PENGUJIAN

Pengujian pada aplikasi navigasi menggunakan kamera sebagai sensor pengambil data gambar ini diimplementasikan menggunakan *mobile robot Qbot*. *Mobile robot Qbot* terhubung dengan komputer sehingga data yang diperoleh robot seperti koordinat jarak dapat ditampilkan pada simulasi dan juga tampilan sirkuitnya. Simulasi secara keseluruhannya adalah kendaraan dapat mengikuti jalan sepanjang sirkuit jalur yang telah disediakan. Pada seluruh pengujiannya sistem navigasi Matlab 2011b yang telah diberi ekstensi *QUARC*.

### 4.1. Implementasi Perencanaan Navigasi

Pengujian navigasi *mobile robot* adalah pengujian kemampuan sistem yang diberikan pada robot untuk mendeteksi jalur yang akan dilalui robot. Titik-titik yang terdeteksi menjadi titik koordinat target yang akan dilalui robot. Jika koordinat lebih kekanan dari titik tengah maka *mobile robot* akan menuju ke kanan, dan sebaliknya jika koordinat yang dideteksi lebih ke kiri maka robot akan bergerak ke kiri. Namun jika koordinat yang dideteksi yang merupakan jalur *mobile robot Qbot* maka robot jalan lurus dan selalu menyesuaikan data koordinat yang akan dideteksi robot sebagai jalur. Titik-titik koordinat yang terdeteksi merupakan obyek yang akan dikenali polanya. Obyek tersebut akan dikenali polanya saat jarak robot dan obyek 15 cm di depan robot. Obyek tersebut merupakan rambu-rambu buat robot apakah rambu-rambu belok kanan, kiri atau stop.

### 4.2. Pengujian Identifikasi Obyek



**Gambar 4.1** Obyek Yang Menjadi Navigasi

Pengujian diambil dari gambar 4.1 akan dipisahkan antara obyek sebagai Navigasi dan bukan obyek dipisahkan dalam bentuk hitam putih

sebelum dijadikan input untuk jaringan saraf tiruan. Maka sesuai dengan elemen dasar gambar untuk pengenalan pola yaitu kekontrasan, kecerahan, warna, bentuk, tekstur, waktu dan pergerakan maka perlu melakukan pemisahan elemen *image* dengan *background* gambar seperti pada gambar 4.2 untuk gambar persegi.



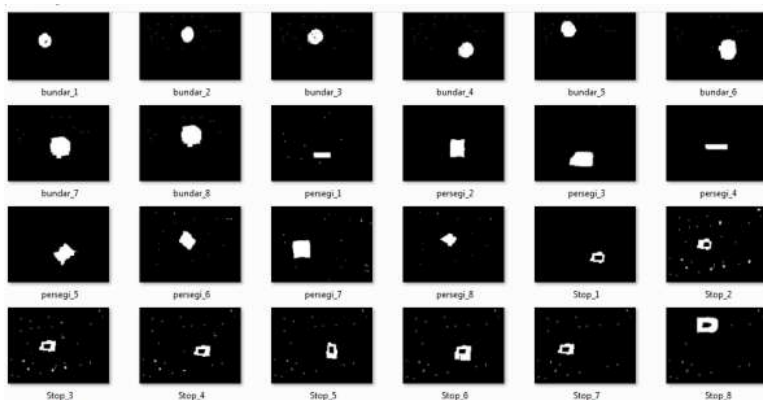
**Gambar 4.2** Gambar Obyek ditampilkan dan *background* dihitamkan

Pada pemrosesan gambar telah terpisahkan antara obyek dan *background*, maka gambar tersebut menjadi nilai *input* pengenalan pola untuk dijadikan navigasi. Kemudian melakukan proses *crop* gambar agar nilai benda semakin mudah dikenal dan cepat diakses. *Pixel crop* menjadi 46x46 *pixel* seperti pada Gambar 4.3.



**Gambar 4.3** Obyek Dalam Bentuk Data Gambar Yang Telah di-*Crop*

Untuk pengujian dimasukan beberapa data gambar, semakin banyak gambar yang dikenali dan memiliki karakteristik yang sama maka proses pengklasifikasian mampu dillakukan dengan baik dan cepat berikut data gambar yang akan dikelasifikasi untuk data *training seperti* pada Gambar 4.4 merupan gambar-gambar yang akan dijadikan input untuk pengenalan pola.



**Gambar 4. 4** Benda-Benda Yang Dijadikan Obyek Navigasi

Berikut hasil pengujian menggunakan metode momenhu maka menghasilkan keluaran seperti Table 4.1 yang nantinya dijadikan data *training*.

**Tabel 4.1** Hasil pengujian pada 3 obyek yang berbeda masing-masing memiliki 7 karakteristik berbeda-beda.

<i>Image</i>	x1	x2	x3	x4	x5	x6	x7
bundar_1	0.16	0.0001	0	0	0	0	0
bundar_2	0.1636	0.0007	0	0	0	0	0
bundar_3	0.1652	0.0003	0	0	0	0	0
bundar_4	0.1621	0.0004	0	0	0	0	0
bundar_5	0.1635	0.0005	0	0	0	0	0
bundar_6	0.1672	0.0019	0	0	0	0	0
bundar_7	0.1684	0.0016	0	0	0	0	0
bundar_8	0.1684	0.0016	0	0	0	0	0
Persegi_1	0.2916	0.0459	0.0003	0.0001	0	0	0
Persegi_2	0.262	0.0414	0	0	0	0	0
Persegi_3	0.2131	0.0179	0	0	0	0	0
Persegi_4	0.2863	0.0546	0	0	0	0	0
Persegi_5	0.3372	0.0832	0.0028	0.0016	0	0.4	0

Persegi _6	0.2815	0.051	0	0	0	0.0004	0
Persegi _7	0.2916	0.0459	0.0003	0.0001	0	0	0
Persegi _8	0.3134	0.0692	0.0008	0.0002	0	0	0
Stop _1	0.1797	0.0007	0.0019	0	0	0	0
Stop _2	0.2034	0.0051	0.0003	0.1	0	0	0
Stop _3	0.1871	0.0003	0.0026	0	0	0	0
Stop _4	0.1878	0.0006	0.0025	0	0	0	0
Stop _5	0.1807	0.0006	0.0019	0	0	0	0
Stop _6	0.1857	0.0006	0.002	0	0	0	0
Stop _7	0.1844	0.0011	0.0019	0	0	0	0
Stop _8	0.1857	0.0006	0.002	0	0	0	0

Maka hasil target *output training* didekati dengan melakukan training pada data input di atas sehingga diperoleh *Output Training* sebagai nilai prediksi navigasi.

**Tabel 4.2** Target training dan keluaran training

Bentuk Image	Target Output	Output Training	Error
bundar_1.jpg	0.5	0.5978	0.0978
bundar_2.jpg	0.5	0.5835	0.0835
bundar_3.jpg	0.5	0.5712	0.0712
bundar_4.jpg	0.5	0.5924	0.0924
bundar_5.jpg	0.5	0.5848	0.0848
bundar_6.jpg	0.5	0.5313	0.0313
bundar_7.jpg	0.5	0.5098	0.0098
bundar_8.jpg	0.5	0.5098	0.0098
Persegi_1.jpg	0.9	0.8978	-0.0022
Persegi_2.jpg	0.9	0.9203	0.0203
Persegi_3.jpg	0.9	0.9094	0.0094
Persegi_4.jpg	0.9	0.8973	-0.0027

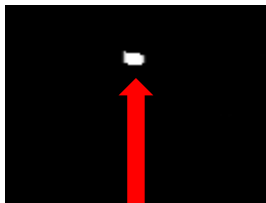
Persegi _5.jpg	0.9	0.883	-0.017
Persegi _6.jpg	0.9	0.9007	0.0007
Persegi _7.jpg	0.9	0.8978	-0.0022
Persegi _8.jpg	0.9	0.8873	-0.0127
Stop_1.jpg	0.1	0.2286	0.1286
Stop_2.jpg	0.1	0.2258	0.1258
Stop _3.jpg	0.1	0.1694	0.0694
Stop _4.jpg	0.1	0.1675	0.0675
Stop _5.jpg	0.1	0.2141	0.1141
Stop _6.jpg	0.1	0.1732	0.0732
Stop _7.jpg	0.1	0.178	0.078
Stop _8.jpg	0.1	0.1732	0.0732

### 4.3. Pengujian Pada Jalur

Pengujian menggunakan kamera dan diterapkan pada *mobile robot Qbot* untuk mendeteksi obyek yang berwarna biru dan selain warna biru dianggap tidak ada obyek.

#### 4.3.1. Pengujian Pada Jalan Lurus dan Robot Jalan Lurus

Pengujian pertama kondisi saat obyek berada di depan bagian posisi tengah kendaraan maka robot akan jalan lurus mendekati obyek hingga jarak 15 cm didepan robot dan berhenti terlihat pada Gambar. 4.5.



**Gambar 4. 5** Pengujian Pada Jalan lurus dan Robot Jalan Lurus

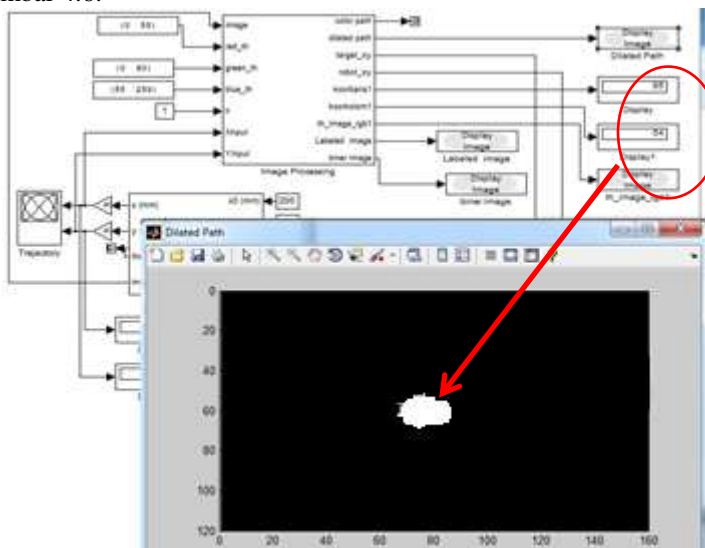
#### 4.3.2 Penempatan Obyek Berada di Sisi Kanan dan Kiri Robot

Pada pengujian ini dimana bagaimana jika obyek di tempatkan pada sisi kanan atau sisi kiri dari robot.

Jika obyek berada di sisi Kanan maka robot berbelok ke kanan hingga obyek berada dititik tengah *pixel* baris hingga jarak robot dan obyek berjarak 15 cm di depan robot maka selanjutnya robot akan benrhenti. Begitu juga kondisinya jika obyek ditempatkan di sebelah kiri depan robot, maka robot belok kiri hingga titik tengah robot berada di titik tengah *pixel* baris kamera robot.

#### 4.3.3 Kondisi Navigasi Jalan Belok Kanan dan Robot Belok Kanan

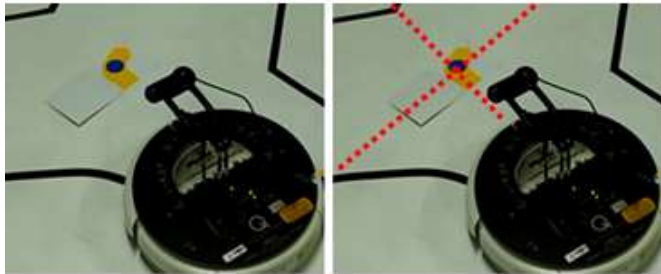
Pengujian ketiga kondisi kendaraan berada di jalur belok kanan maka mobile robot akan bergerak belok ke kanan. Data terlihat pada Gambar 4.6.



**Gambar 4. 6** Kondisi Jalan Belok Kanan dan Robot Belok Kanan

Untuk gambar sebenarnya seperti pada Gambar 4.7 berikut dimana untuk ditengah tanda panah kiri yang berwarna kuning terdapat bundar berwarna biru sebagai bentuk gambar yang dikenali robot melalui *training* jaringan saraf tiruan sebelum pengujiannya.



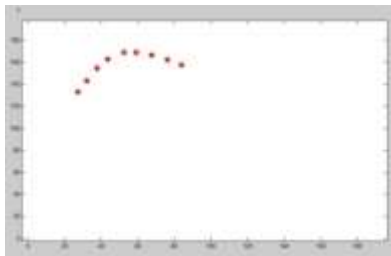


a.

b.

**Gambar 4. 7** a.Gambar Jalur Jika Dilihat Dari Atas Robot b.Koordinat Obyek Terhadap Kamera

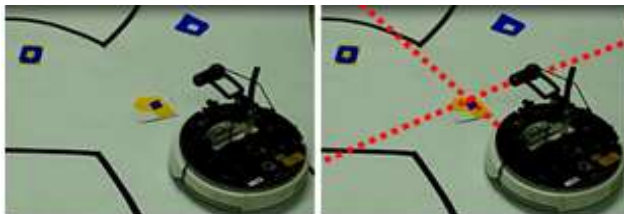
Hasil ditampilkan pada Trajectory diperoleh hasil sebagai berikut [1] Gambar 4.8.



**Gambar 4. 8** Tampilan trajectory pada saat belok kanan

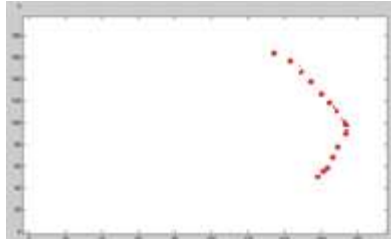
#### 4.3.3 Kondisi Navigasi Jalan Belok Kiri dan Robot Belok Kiri

Pengujian ketika *mobile robot* jalur belok kiri dari posisi *mobile robot Qbot* maka robot akan berbeok ke kiri seperti pada Gambar 4.9.



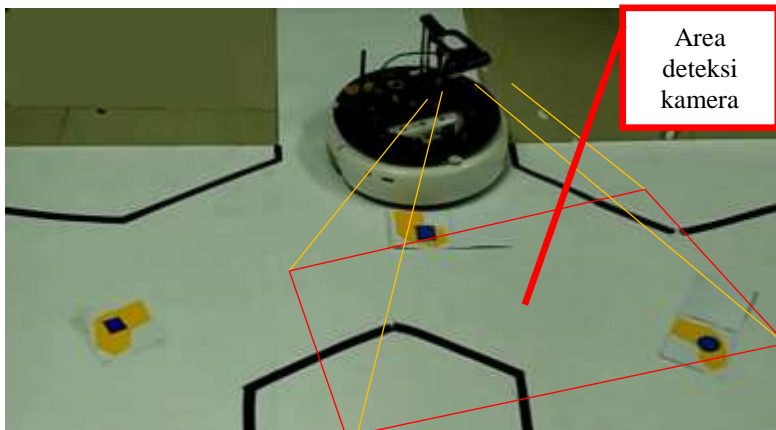
**Gambar 4. 9** Kondisi Navigasi Belok Kiri dan Robot Belok Kiri

Perjalanan robot pada jalurnya ditampilkan pada *plot trajectory*. Pada hasil percobaan pada Gambar 4.10 hanya menampilkan pergerakan belok kiri.



**Gambar 4. 10** Trajectory yang ditampilkan Saat Belok Kiri

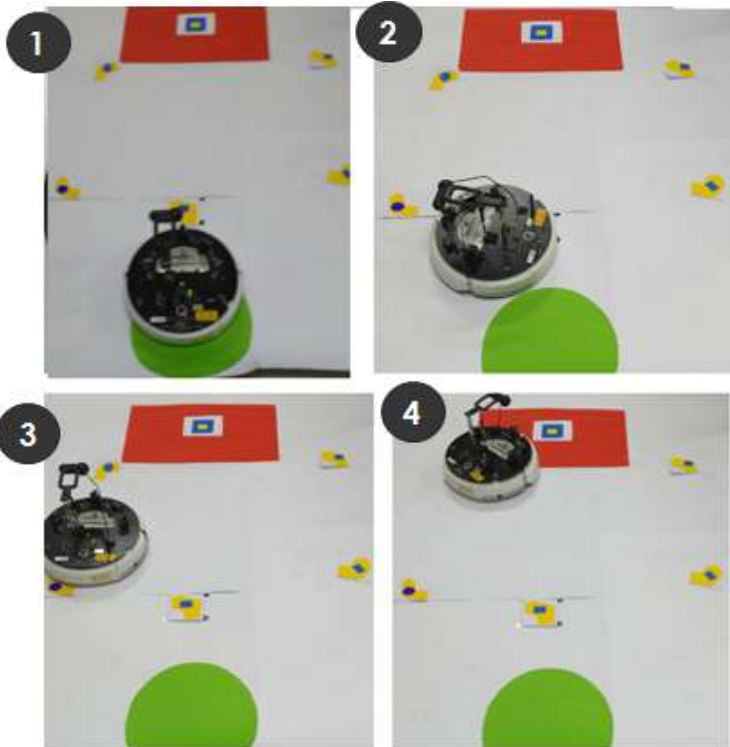
#### 4.3.4 Pengujian Pada *Track 1*



**Gambar 4. 11** Area Yang Masuk ke Dalam *Capture Picture*

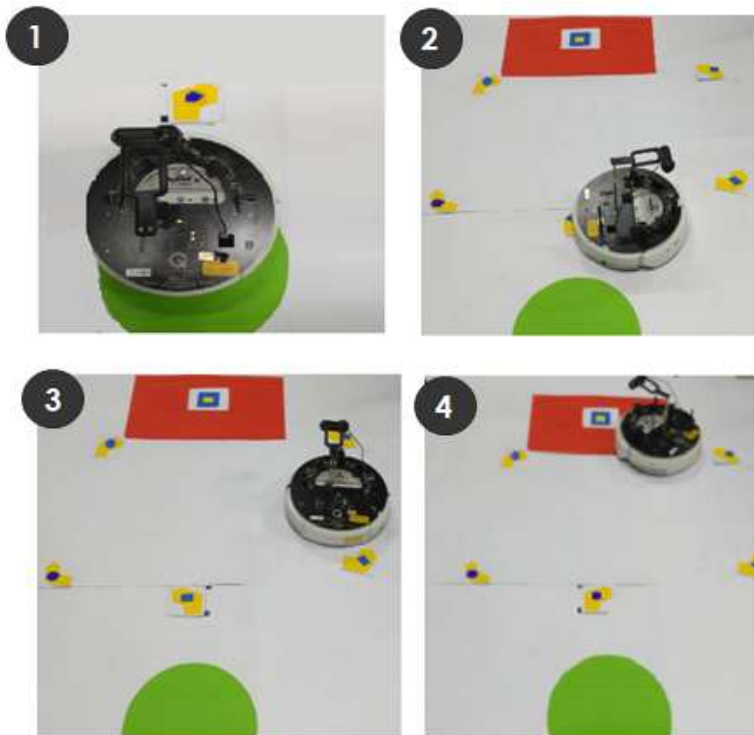
#### 4.3.6 Pengujian Pada Track 2

Pengujian pada track 2 dengan memberikan navigasi belok kiri di pertigaan track, kemudian *mobile robot* Qbot di operasikan seperti pada Gambar 4.12.



Gambar 4.12 Pengujian Pada Track 2 Saat pertigaan Belok Kiri

Sebagai pembandingannya pengujian pada track 2 dengan memberikan navigasi belok kanan di pertigaan track, kemudian *mobile robot* Qbot di operasikan seperti pada Gambar 4.13.



Gambar 4.12 Pengujian Pada *Track 2* Saat pertigaan Belok Kanan

#### 4.4 Pengukuran cahaya

Pengukuran cahaya pada *track mobile robot* dioperasikan tujuannya untuk mengetahui bahwa kemampuan robot memiliki keterbatasan dalam cahaya yang redup. Pengukuran menggunakan luxmeter.

Pengukuran intensitas cahaya di Laboratorium Sistem dan Sibernetika dengan menghidupkan 2 lampu neon pada Pukul 14:30 WIB seperti pada pengukuran seperti pada Gambar 4.13 berikut.



(a)



(b)

Gambar 4.13 Pengukuran Intensitas Cahaya Dengan Menghidupkan 2 Lampu Neon Dalam Ruangan

Terukur:

Intensitas Cahaya : 244 lux

Pengukuran selanjutnya mengurangi intensitas cahaya dengan menghidupkan 2 lampu neon Pada Pukul 14:30 WIB



(a)



(b)

Gambar 4.14 a. Pengukuran Intensitas Cahaya Menghidupkan 1 Lampu Neon Dalam Ruangan b. Tampilan Gambar

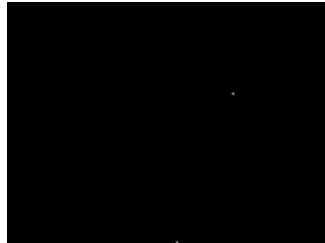
Terukur:

Intensitas Cahaya : 193 lux

Pengukuran selanjutnya mengurangi intensitas cahaya dengan mematikan semua neon Pada Pukul 14:30 WIB



(a)



(b)

Gambar 4.15 a. Pengukuran Intensitas Cahaya Dengan Mematikan Semua Lampu Neon pada Pukul 14:30 WIB b. Tampilan Gambar

Terukur:

Intensitas Cahaya : 30 lux

Pengukuran dengan menambahkan lampu senter *Mobile Phone Xiami red mi-3* dan dua lampu neon hidup.



(a)



(b)

Gambar 4.16 a. Pengukuran Intensitas Cahaya Dengan Menghidupkan 3 Lampu Neon dan menambahkan dengan Lampu Senter *Mobile Phone Xiami red mi-3* b. Tampilan Gambar

Terukur:

Intensitas Cahaya : 331 lux

## **BAB V**

### **KESIMPULAN DAN SARAN**

#### **5.1. Kesimpulan**

Berdasarkan data hasil pengujian dalam pembuatan tugas akhir ini, dapat ditarik kesimpulan sebagai berikut:

1. Sistem navigasi yang diterapkan pada mobile robot pada jalur yang ditentukan merupakan pengambilan titik target yang akan dituju, disetiap ada kondisi maka mempertimbangkan arah tujuan. Sistem ini dihadapkan beberapa pilihan seperti rambu-rambu belok kanan, belok kiri dan berhenti layaknya navigasi yang menjadi penunjuk arah pada jalur sehingga *mobile robot* mampu mengikuti jalur tersebut.
2. Pada hasil tampilan *trajectory* terdapat perbedaan kordinat dengan tampilan pixel, yaitu koordinat pixel X,Y(1,1) berada pada kiri atas dimensi dua sedangkan pada koordinat area untuk nilai koordinat X,Y (0,0) berada pada kiri bawah sehingga mempengaruhi dan gerak lurus menjadi menjauhi titik awal 0,0 ke arah susut 45 derajat sedangkan untuk kamera robot berjalan lurus ke depan 90 derajat dalam perbandingan *pixel* dengan posisi koordinat robot.
3. Kemampuan robot dalam mengenali pola cukup baik karena saat diuji dengan navigasi, robot mampu berbelok ke kanan, belok kiri atau berhenti sesuai navigasi.

#### **5.2. Saran**

Saran yang dapat diberikan oleh penulis untuk pengembangan dari tugas akhir ini yaitu sebagai berikut:

Pengembangan untuk sistem seperti ini dibutuhkan proses yang cepat untuk data kamera sementara untuk akses data gambar *processor* robot hanya mampu mengakses 9 *frame* per-detik sehingga gambar yang disajikan sangat lamban dan susah untuk mendeteksi benda saat bergerak disebabkan data aksesnya terlambat untuk frame saat ini untuk kenyataan untuk *Image processing* minimal 45 *frame* per-detik.

*Halaman ini sengaja dikosongkan*



## DAFTAR PUSTAKA

- [1] \_\_\_\_\_, *User Manual Quanser Qbot*, 7th ed., Document Numbe 830, Ed.: Quanser, 2012.
- [2] C. Gonzales Rafael and E. Woods Richard, *Diital Image Processing*, 2nd ed., Marcial J. Horton, Ed. Saddle River, New Jarsey: Tom Robbins, 2001.
- [3] Suyanto, *Artificial Intelligence*. Bandung, Indonesia: Informatika, 2014.
- [4] Laurene Fausett, *Fundamentals Of Neural Networks :Architecture, Algorithm and Application*. Saddle River, USA: Addison-Wesley Publishing Company, Inc., 1994.
- [5] Dong Ping Tian, "A Review on *Image* Feature Extraction and Representation Techniques," *International Journal of Multimedia and Ubiquitous Engineering*, vol. 8, p. 12, July 2013.
- [6] Abdul Khadir and Adhi Susanto, *Pengolahan Citra Teori dan Aplikasi*, 1st ed., Dewibertha Hardjono, Ed. Yogyakarta , Indonesia : Andi, 2013.

*Halaman ini sengaja dikosongkan*

## LAMPIRAN

### Lampiran A.1 Program *Crop* Gambar

```
Function Crop = crop_obyek (G)
[m,n]= size(G);
mm= m/2; % 1/2 dari pixel baris
BARIS=0;
KOLOM=0;
putihkolom=zeros(1,m);
putihbaris=zeros(1,n);
lebarbaris=0;
nilaikolom=0;
jlm_adanilai_baris_atas=0;
jlm_adanilai_baris_bawah=0;
jlm_adanilai_baris=0;
jlm_adanilaikolom=0;
jlm_adanilaikolom_kanan=0;
jlm_adanilaikolom_kiri=0;

b=0;
a=0;
sw=0;
    yw=1;
    kkm=m-1;
    for i=1:kkm %program mendeteksi koor Y
        ketemuputihkolom=0;
        batasKet=m-1;
        for j=1:batasKet
            kkm=n-1;
            if G(j,i)==255
                ketemuputihkolom=1;
                putihkolom(i)=putihkolom(i)+1;
            end
            if G(j,i)==0 && ketemuputihkolom==1
                break;
            end
        end
    end
    batasKet1=m-1;
    for i=1:batasKet1 %program mendeteksi koor x
        ketemuputihbaris=0;
        batasKet3=n-1;
```

```

for j=1:batasKet3
    if G(i,j)==255
        ketemuputihbaris=1;
        putihbaris(i)=putihbaris(i)+1;
    end
    if G(i,j)==0 && ketemuputihbaris==1
        break;
    end
end
end
AC=putihkolom>=max(putihkolom);
BK=n-1;
for i=1:BK
    if(AC(1,i)==1)
        KOLOM=i;
        break;
    end
end

KOLOM;
AB=putihbaris>=max(putihbaris);
BM=m-1;
for i=1:BM
    if(AB(1,i)==1)
        BARIS=i;
        break;
    end
end

%untuk beberapa matlab bisa menggunakan skript berikut
% % % AB=find(putihbaris>=max(putihbaris));
% % % max(AB)
JARAK=n-BARIS;
BARIS;
%LGORITHMA ITERASI KE bawah
batasbawah=m-BARIS;
for i=1:batasbawah
    if G(BARIS+i,KOLOM)>=100
        A=G(BARIS+i,KOLOM);
        jlm_adanilai_baris_bawah=jlm_adanilai_baris_bawah+1;
    end
end
% %ALGORITHMMA ITERASI KE atas
batas_atas1=BARIS-1;

```

```

for i=1:batas_atas1
    if G(BARIS-i,KOLOM)>=100
        A=G(BARIS-i,KOLOM);
        jlm_adanilai_baris_atas=jlm_adanilai_baris_atas+1;
    end
end

%LGORITHMA ITERASI KE KANAN
bataskolomkanan=m-KOLOM;
for i=1:bataskolomkanan
    if G(BARIS,KOLOM+i)>=100
        B=G(BARIS,KOLOM+i);
        jlm_adanilaikolom_kanan=jlm_adanilaikolom_kanan+1;
    end
end
%LGORITHMA ITERASI KE KIRI
bataskolomkiri=KOLOM-1;
for i=1:bataskolomkiri
    if G(BARIS,KOLOM-i)>=100
        B=G(BARIS,KOLOM-i);
        jlm_adanilaikolom_kiri=jlm_adanilaikolom_kiri+1;
    end
end
jlm_adanilai_baris_atas;
jlm_adanilai_baris_bawah;
jlm_adanilaikolom_kanan;
jlm_adanilaikolom_kiri;
jlm_adanilai_baris=jlm_adanilai_baris_atas+jlm_adanilai_baris_bawah;
jlm_adanilaikolom=jlm_adanilaikolom_kanan-jlm_adanilaikolom_kiri;
TotalBaris=jlm_adanilai_baris_atas+jlm_adanilai_baris_bawah;

Tengah_baris=floor(TotalBaris/2); %floor pembulatan angka agar tidak
berkoma
TengahBaris=Tengah_baris-jlm_adanilai_baris_bawah; %patokan dari
bawah sekali-koordinatsekarang

TotalKolom=jlm_adanilaikolom_kanan-jlm_adanilaikolom_kiri;
Tengah_kolom=floor(TotalKolom/2);
KOLOM=KOLOM+Tengah_kolom;

Asz=[
G(BARIS-TengahBaris-23,KOLOM-21)...

```

G(BARIS-TengahBaris-23,KOLOM-20)	G(BARIS-TengahBaris-23,KOLOM-19)
G(BARIS-TengahBaris-23,KOLOM-17)	G(BARIS-TengahBaris-23,KOLOM-16)
G(BARIS-TengahBaris-23,KOLOM-14)	G(BARIS-TengahBaris-23,KOLOM-13)
G(BARIS-TengahBaris-23,KOLOM-11)	G(BARIS-TengahBaris-23,KOLOM-10)
G(BARIS-TengahBaris-23,KOLOM-9)	G(BARIS-TengahBaris-23,KOLOM-8)
G(BARIS-TengahBaris-23,KOLOM-7)	G(BARIS-TengahBaris-23,KOLOM-6)
G(BARIS-TengahBaris-23,KOLOM-5)	G(BARIS-TengahBaris-23,KOLOM-4)
G(BARIS-TengahBaris-23,KOLOM-3)	G(BARIS-TengahBaris-23,KOLOM-2)
G(BARIS-TengahBaris-23,KOLOM-0)	G(BARIS-TengahBaris-23,KOLOM+1)
G(BARIS-TengahBaris-23,KOLOM+2)	G(BARIS-TengahBaris-23,KOLOM+3)
G(BARIS-TengahBaris-23,KOLOM+4)	G(BARIS-TengahBaris-23,KOLOM+5)
G(BARIS-TengahBaris-23,KOLOM+6)	G(BARIS-TengahBaris-23,KOLOM+7)
G(BARIS-TengahBaris-23,KOLOM+9)	G(BARIS-TengahBaris-23,KOLOM+10)
G(BARIS-TengahBaris-23,KOLOM+11)	G(BARIS-TengahBaris-23,KOLOM+12)
G(BARIS-TengahBaris-23,KOLOM+14)	G(BARIS-TengahBaris-23,KOLOM+15)
G(BARIS-TengahBaris-23,KOLOM+17)	G(BARIS-TengahBaris-23,KOLOM+18)
G(BARIS-TengahBaris-23,KOLOM+20)	G(BARIS-TengahBaris-23,KOLOM+21)
G(BARIS-TengahBaris-22,KOLOM-21)	G(BARIS-TengahBaris-22,KOLOM-19)
G(BARIS-TengahBaris-22,KOLOM-17)	G(BARIS-TengahBaris-22,KOLOM-16)
G(BARIS-TengahBaris-22,KOLOM-14)	G(BARIS-TengahBaris-22,KOLOM-13)
G(BARIS-TengahBaris-22,KOLOM-11)	G(BARIS-TengahBaris-22,KOLOM-10)

G(BARIS-TengahBaris-22,KOLOM-8)	G(BARIS-TengahBaris-22,KOLOM-7) G(BARIS-TengahBaris-22,KOLOM-6)...
G(BARIS-TengahBaris-22,KOLOM-5)	G(BARIS-TengahBaris-22,KOLOM-4)...
G(BARIS-TengahBaris-22,KOLOM-3)	G(BARIS-TengahBaris-22,KOLOM-2) G(BARIS-TengahBaris-22,KOLOM-1) ...
G(BARIS-TengahBaris-22,KOLOM-0)	G(BARIS-TengahBaris-22,KOLOM+1) G(BARIS-TengahBaris-22,KOLOM+2) ...
G(BARIS-TengahBaris-22,KOLOM+2)	G(BARIS-TengahBaris-22,KOLOM+3) G(BARIS-TengahBaris-22,KOLOM+3) ...
G(BARIS-TengahBaris-22,KOLOM+4)	G(BARIS-TengahBaris-22,KOLOM+5) ...
G(BARIS-TengahBaris-22,KOLOM+6)	G(BARIS-TengahBaris-22,KOLOM+7) G(BARIS-TengahBaris-22,KOLOM+8)...
G(BARIS-TengahBaris-22,KOLOM+9)	G(BARIS-TengahBaris-22,KOLOM+10)...
G(BARIS-TengahBaris-22,KOLOM+11)	G(BARIS-TengahBaris-22,KOLOM+12) G(BARIS-TengahBaris-22,KOLOM+13)...
G(BARIS-TengahBaris-22,KOLOM+14)	G(BARIS-TengahBaris-22,KOLOM+15) G(BARIS-TengahBaris-22,KOLOM+16)...
G(BARIS-TengahBaris-22,KOLOM+17)	G(BARIS-TengahBaris-22,KOLOM+18) G(BARIS-TengahBaris-22,KOLOM+19)...
G(BARIS-TengahBaris-22,KOLOM+20)	G(BARIS-TengahBaris-22,KOLOM+21) G(BARIS-TengahBaris-22,KOLOM+22);
G(BARIS-TengahBaris-21,KOLOM-21)...	
G(BARIS-TengahBaris-21,KOLOM-20)	G(BARIS-TengahBaris-21,KOLOM-19) G(BARIS-TengahBaris-21,KOLOM-18)...
G(BARIS-TengahBaris-21,KOLOM-17)	G(BARIS-TengahBaris-21,KOLOM-16) G(BARIS-TengahBaris-21,KOLOM-15)...
G(BARIS-TengahBaris-21,KOLOM-14)	G(BARIS-TengahBaris-21,KOLOM-13) G(BARIS-TengahBaris-21,KOLOM-12)...
G(BARIS-TengahBaris-21,KOLOM-11)	G(BARIS-TengahBaris-21,KOLOM-10) G(BARIS-TengahBaris-21,KOLOM-9)...
G(BARIS-TengahBaris-21,KOLOM-8)	G(BARIS-TengahBaris-21,KOLOM-7) G(BARIS-TengahBaris-21,KOLOM-6)...
G(BARIS-TengahBaris-21,KOLOM-4)	G(BARIS-TengahBaris-21,KOLOM-3) G(BARIS-TengahBaris-21,KOLOM-2)...
G(BARIS-TengahBaris-21,KOLOM-1)	G(BARIS-TengahBaris-21,KOLOM-0) G(BARIS-TengahBaris-21,KOLOM+1)...
G(BARIS-TengahBaris-21,KOLOM+2)	G(BARIS-TengahBaris-21,KOLOM+2) G(BARIS-TengahBaris-21,KOLOM+3)...
G(BARIS-TengahBaris-21,KOLOM+3)	G(BARIS-TengahBaris-21,KOLOM+3)

21,KOLOM+4) G(BARIS-TengahBaris-21,KOLOM+4)...  
     G(BARIS-TengahBaris-21,KOLOM+5)                      G(BARIS-TengahBaris-  
 21,KOLOM+6) G(BARIS-TengahBaris-21,KOLOM+7)...  
     G(BARIS-TengahBaris-21,KOLOM+8)                      G(BARIS-TengahBaris-  
 21,KOLOM+9) G(BARIS-TengahBaris-21,KOLOM+10)...  
     G(BARIS-TengahBaris-21,KOLOM+11)                      G(BARIS-TengahBaris-  
 21,KOLOM+12) G(BARIS-TengahBaris-21,KOLOM+13)...  
     G(BARIS-TengahBaris-21,KOLOM+14)                      G(BARIS-TengahBaris-  
 21,KOLOM+15) G(BARIS-TengahBaris-21,KOLOM+16)...  
     G(BARIS-TengahBaris-21,KOLOM+17)                      G(BARIS-TengahBaris-  
 21,KOLOM+18) G(BARIS-TengahBaris-21,KOLOM+19)...  
     G(BARIS-TengahBaris-21,KOLOM+20)                      G(BARIS-TengahBaris-  
 21,KOLOM+21) G(BARIS-TengahBaris-21,KOLOM+22);  
  
     G(BARIS-TengahBaris-20,KOLOM-21)...  
     G(BARIS-TengahBaris-20,KOLOM-20)                      G(BARIS-TengahBaris-  
 20,KOLOM-19) G(BARIS-TengahBaris-20,KOLOM-18)...  
     G(BARIS-TengahBaris-20,KOLOM-17)                      G(BARIS-TengahBaris-  
 20,KOLOM-16) G(BARIS-TengahBaris-20,KOLOM-15)...  
     G(BARIS-TengahBaris-20,KOLOM-14)                      G(BARIS-TengahBaris-  
 20,KOLOM-13) G(BARIS-TengahBaris-20,KOLOM-12)...  
     G(BARIS-TengahBaris-20,KOLOM-11)                      G(BARIS-TengahBaris-  
 20,KOLOM-10) G(BARIS-TengahBaris-20,KOLOM-9)...  
     G(BARIS-TengahBaris-20,KOLOM-8)                      G(BARIS-TengahBaris-  
 20,KOLOM-7) G(BARIS-TengahBaris-20,KOLOM-6)...  
     G(BARIS-TengahBaris-20,KOLOM-4)...  
     G(BARIS-TengahBaris-20,KOLOM-3)                      G(BARIS-TengahBaris-  
 20,KOLOM-2) G(BARIS-TengahBaris-20,KOLOM-1) ...  
     G(BARIS-TengahBaris-20,KOLOM-0)                      G(BARIS-TengahBaris-  
 20,KOLOM+1) G(BARIS-TengahBaris-20,KOLOM+2) ...  
     G(BARIS-TengahBaris-20,KOLOM+2)                      G(BARIS-TengahBaris-  
 20,KOLOM+3) G(BARIS-TengahBaris-20,KOLOM+3) ...  
     G(BARIS-TengahBaris-20,KOLOM+4)                      G(BARIS-TengahBaris-  
 20,KOLOM+4) G(BARIS-TengahBaris-20,KOLOM+5) ...  
     G(BARIS-TengahBaris-20,KOLOM+6)                      G(BARIS-TengahBaris-  
 20,KOLOM+7) G(BARIS-TengahBaris-20,KOLOM+8)...  
     G(BARIS-TengahBaris-20,KOLOM+9)                      G(BARIS-TengahBaris-  
 20,KOLOM+10) G(BARIS-TengahBaris-20,KOLOM+11)...  
     G(BARIS-TengahBaris-20,KOLOM+12)                      G(BARIS-TengahBaris-  
 20,KOLOM+13) G(BARIS-TengahBaris-20,KOLOM+14)...  
     G(BARIS-TengahBaris-20,KOLOM+15)                      G(BARIS-TengahBaris-  
 20,KOLOM+16) G(BARIS-TengahBaris-20,KOLOM+17)...  
     G(BARIS-TengahBaris-20,KOLOM+18)                      G(BARIS-TengahBaris-



20,KOLOM+19) G(BARIS-TengahBaris-20,KOLOM+20)...	
G(BARIS-TengahBaris-20,KOLOM+21)	G(BARIS-TengahBaris-20,KOLOM+22);
G(BARIS-TengahBaris-19,KOLOM-21)...	
G(BARIS-TengahBaris-19,KOLOM-20)	G(BARIS-TengahBaris-19,KOLOM-19) G(BARIS-TengahBaris-19,KOLOM-18)...
G(BARIS-TengahBaris-19,KOLOM-17)	G(BARIS-TengahBaris-19,KOLOM-16) G(BARIS-TengahBaris-19,KOLOM-15)...
G(BARIS-TengahBaris-19,KOLOM-14)	G(BARIS-TengahBaris-19,KOLOM-13) G(BARIS-TengahBaris-19,KOLOM-12)...
G(BARIS-TengahBaris-19,KOLOM-11)	G(BARIS-TengahBaris-19,KOLOM-10) G(BARIS-TengahBaris-19,KOLOM-9)...
G(BARIS-TengahBaris-19,KOLOM-8)	G(BARIS-TengahBaris-19,KOLOM-7) G(BARIS-TengahBaris-19,KOLOM-6)...
G(BARIS-TengahBaris-19,KOLOM-4)...	
G(BARIS-TengahBaris-19,KOLOM-3)	G(BARIS-TengahBaris-19,KOLOM-2) G(BARIS-TengahBaris-19,KOLOM-1) ...
G(BARIS-TengahBaris-19,KOLOM-0)	G(BARIS-TengahBaris-19,KOLOM+1) G(BARIS-TengahBaris-19,KOLOM+2) ...
G(BARIS-TengahBaris-19,KOLOM+2)	G(BARIS-TengahBaris-19,KOLOM+3) G(BARIS-TengahBaris-19,KOLOM+3) ...
G(BARIS-TengahBaris-19,KOLOM+4)	G(BARIS-TengahBaris-19,KOLOM+4) G(BARIS-TengahBaris-19,KOLOM+5) ...
G(BARIS-TengahBaris-19,KOLOM+6)	G(BARIS-TengahBaris-19,KOLOM+7) G(BARIS-TengahBaris-19,KOLOM+8)...
G(BARIS-TengahBaris-19,KOLOM+9)	G(BARIS-TengahBaris-19,KOLOM+10) G(BARIS-TengahBaris-19,KOLOM+11)...
G(BARIS-TengahBaris-19,KOLOM+12)	G(BARIS-TengahBaris-19,KOLOM+13) G(BARIS-TengahBaris-19,KOLOM+14)...
G(BARIS-TengahBaris-19,KOLOM+15)	G(BARIS-TengahBaris-19,KOLOM+16) G(BARIS-TengahBaris-19,KOLOM+17)...
G(BARIS-TengahBaris-19,KOLOM+18)	G(BARIS-TengahBaris-19,KOLOM+19) G(BARIS-TengahBaris-19,KOLOM+20)...
G(BARIS-TengahBaris-19,KOLOM+21)	G(BARIS-TengahBaris-19,KOLOM+22);
G(BARIS-TengahBaris-18,KOLOM-21)...	
G(BARIS-TengahBaris-18,KOLOM-20)	G(BARIS-TengahBaris-18,KOLOM-19) G(BARIS-TengahBaris-18,KOLOM-18)...
G(BARIS-TengahBaris-18,KOLOM-17)	G(BARIS-TengahBaris-18,KOLOM-16) G(BARIS-TengahBaris-18,KOLOM-15)...
G(BARIS-TengahBaris-18,KOLOM-14)	G(BARIS-TengahBaris-18,KOLOM-13) G(BARIS-TengahBaris-18,KOLOM-12)...

18,KOLOM-13) G(BARIS-TengahBaris-18,KOLOM-12)...	
G(BARIS-TengahBaris-18,KOLOM-11)	G(BARIS-TengahBaris-18,KOLOM-10) G(BARIS-TengahBaris-18,KOLOM-9)...
G(BARIS-TengahBaris-18,KOLOM-8)	G(BARIS-TengahBaris-18,KOLOM-7) G(BARIS-TengahBaris-18,KOLOM-6)...
G(BARIS-TengahBaris-18,KOLOM-4)...	G(BARIS-TengahBaris-18,KOLOM-3)
G(BARIS-TengahBaris-18,KOLOM-1) ...	G(BARIS-TengahBaris-18,KOLOM-0)
G(BARIS-TengahBaris-18,KOLOM+1) G(BARIS-TengahBaris-18,KOLOM+2) ...	G(BARIS-TengahBaris-18,KOLOM+3) G(BARIS-TengahBaris-18,KOLOM+4)
G(BARIS-TengahBaris-18,KOLOM+2)	G(BARIS-TengahBaris-18,KOLOM+5) ...
G(BARIS-TengahBaris-18,KOLOM+3)	G(BARIS-TengahBaris-18,KOLOM+6)
G(BARIS-TengahBaris-18,KOLOM+4)	G(BARIS-TengahBaris-18,KOLOM+7) G(BARIS-TengahBaris-18,KOLOM+8)...
G(BARIS-TengahBaris-18,KOLOM+5)	G(BARIS-TengahBaris-18,KOLOM+9)
G(BARIS-TengahBaris-18,KOLOM+6)	G(BARIS-TengahBaris-18,KOLOM+10) G(BARIS-TengahBaris-18,KOLOM+11)...
G(BARIS-TengahBaris-18,KOLOM+7)	G(BARIS-TengahBaris-18,KOLOM+12)
G(BARIS-TengahBaris-18,KOLOM+8)	G(BARIS-TengahBaris-18,KOLOM+13) G(BARIS-TengahBaris-18,KOLOM+14)...
G(BARIS-TengahBaris-18,KOLOM+9)	G(BARIS-TengahBaris-18,KOLOM+15)
G(BARIS-TengahBaris-18,KOLOM+10)	G(BARIS-TengahBaris-18,KOLOM+16) G(BARIS-TengahBaris-18,KOLOM+17)...
G(BARIS-TengahBaris-18,KOLOM+11)	G(BARIS-TengahBaris-18,KOLOM+18)
G(BARIS-TengahBaris-18,KOLOM+12)	G(BARIS-TengahBaris-18,KOLOM+19) G(BARIS-TengahBaris-18,KOLOM+20)...
G(BARIS-TengahBaris-18,KOLOM+13)	G(BARIS-TengahBaris-18,KOLOM+21)
G(BARIS-TengahBaris-18,KOLOM+14)	G(BARIS-TengahBaris-18,KOLOM+22);
G(BARIS-TengahBaris-17,KOLOM-21)...	
G(BARIS-TengahBaris-17,KOLOM-20)	G(BARIS-TengahBaris-17,KOLOM-19) G(BARIS-TengahBaris-17,KOLOM-18)...
G(BARIS-TengahBaris-17,KOLOM-17)	G(BARIS-TengahBaris-17,KOLOM-16) G(BARIS-TengahBaris-17,KOLOM-15)...
G(BARIS-TengahBaris-17,KOLOM-14)	G(BARIS-TengahBaris-17,KOLOM-13) G(BARIS-TengahBaris-17,KOLOM-12)...
G(BARIS-TengahBaris-17,KOLOM-11)	G(BARIS-TengahBaris-17,KOLOM-10) G(BARIS-TengahBaris-17,KOLOM-9)...
G(BARIS-TengahBaris-17,KOLOM-8)	G(BARIS-TengahBaris-17,KOLOM-7) G(BARIS-TengahBaris-17,KOLOM-6)...
G(BARIS-TengahBaris-17,KOLOM-4)...	G(BARIS-TengahBaris-17,KOLOM-3)
G(BARIS-TengahBaris-17,KOLOM-1) ...	G(BARIS-TengahBaris-17,KOLOM-0)
G(BARIS-TengahBaris-17,KOLOM+1) G(BARIS-TengahBaris-17,KOLOM+2) ...	



G(BARIS-TengahBaris-16,KOLOM+15)	G(BARIS-TengahBaris-16,KOLOM+16)
G(BARIS-TengahBaris-16,KOLOM+17)...	G(BARIS-TengahBaris-16,KOLOM+18)
G(BARIS-TengahBaris-16,KOLOM+19)	G(BARIS-TengahBaris-16,KOLOM+20)...
G(BARIS-TengahBaris-16,KOLOM+21)	G(BARIS-TengahBaris-16,KOLOM+22);
G(BARIS-TengahBaris-15,KOLOM-21)...	G(BARIS-TengahBaris-15,KOLOM-20)
G(BARIS-TengahBaris-15,KOLOM-19)	G(BARIS-TengahBaris-15,KOLOM-18)...
G(BARIS-TengahBaris-15,KOLOM-17)	G(BARIS-TengahBaris-15,KOLOM-16)
G(BARIS-TengahBaris-15,KOLOM-15)...	G(BARIS-TengahBaris-15,KOLOM-14)
G(BARIS-TengahBaris-15,KOLOM-13)	G(BARIS-TengahBaris-15,KOLOM-12)...
G(BARIS-TengahBaris-15,KOLOM-11)	G(BARIS-TengahBaris-15,KOLOM-10)
G(BARIS-TengahBaris-15,KOLOM-9)...	G(BARIS-TengahBaris-15,KOLOM-8)
G(BARIS-TengahBaris-15,KOLOM-7)	G(BARIS-TengahBaris-15,KOLOM-6)...
G(BARIS-TengahBaris-15,KOLOM-4)...	G(BARIS-TengahBaris-15,KOLOM-3)
G(BARIS-TengahBaris-15,KOLOM-2)	G(BARIS-TengahBaris-15,KOLOM-1) ...
G(BARIS-TengahBaris-15,KOLOM-0)	G(BARIS-TengahBaris-15,KOLOM+1)
G(BARIS-TengahBaris-15,KOLOM+2) ...	G(BARIS-TengahBaris-15,KOLOM+3)
G(BARIS-TengahBaris-15,KOLOM+4)	G(BARIS-TengahBaris-15,KOLOM+5) ...
G(BARIS-TengahBaris-15,KOLOM+6)	G(BARIS-TengahBaris-15,KOLOM+7)
G(BARIS-TengahBaris-15,KOLOM+8)...	G(BARIS-TengahBaris-15,KOLOM+9)
G(BARIS-TengahBaris-15,KOLOM+10)	G(BARIS-TengahBaris-15,KOLOM+11)...
G(BARIS-TengahBaris-15,KOLOM+12)	G(BARIS-TengahBaris-15,KOLOM+13)
G(BARIS-TengahBaris-15,KOLOM+14)...	G(BARIS-TengahBaris-15,KOLOM+15)
G(BARIS-TengahBaris-15,KOLOM+16)	G(BARIS-TengahBaris-15,KOLOM+17)...
G(BARIS-TengahBaris-15,KOLOM+18)	G(BARIS-TengahBaris-15,KOLOM+19)
G(BARIS-TengahBaris-15,KOLOM+20)...	G(BARIS-TengahBaris-15,KOLOM+21)
G(BARIS-TengahBaris-15,KOLOM+22);	G(BARIS-TengahBaris-14,KOLOM-21)...
G(BARIS-TengahBaris-14,KOLOM-20)	G(BARIS-TengahBaris-14,KOLOM-19)
G(BARIS-TengahBaris-14,KOLOM-18)...	

G(BARIS-TengahBaris-14,KOLOM-17)	G(BARIS-TengahBaris-14,KOLOM-16)	G(BARIS-TengahBaris-14,KOLOM-15)...
G(BARIS-TengahBaris-14,KOLOM-14)	G(BARIS-TengahBaris-14,KOLOM-13)	G(BARIS-TengahBaris-14,KOLOM-12)...
G(BARIS-TengahBaris-14,KOLOM-11)	G(BARIS-TengahBaris-14,KOLOM-10)	G(BARIS-TengahBaris-14,KOLOM-9)...
G(BARIS-TengahBaris-14,KOLOM-8)	G(BARIS-TengahBaris-14,KOLOM-7)	G(BARIS-TengahBaris-14,KOLOM-6)...
G(BARIS-TengahBaris-14,KOLOM-4)...	G(BARIS-TengahBaris-14,KOLOM-3)	G(BARIS-TengahBaris-14,KOLOM-2)
G(BARIS-TengahBaris-14,KOLOM-3)	G(BARIS-TengahBaris-14,KOLOM-2)	G(BARIS-TengahBaris-14,KOLOM-1) ...
G(BARIS-TengahBaris-14,KOLOM-0)	G(BARIS-TengahBaris-14,KOLOM+1)	G(BARIS-TengahBaris-14,KOLOM+2) ...
G(BARIS-TengahBaris-14,KOLOM+2)	G(BARIS-TengahBaris-14,KOLOM+3)	G(BARIS-TengahBaris-14,KOLOM+3) ...
G(BARIS-TengahBaris-14,KOLOM+4)	G(BARIS-TengahBaris-14,KOLOM+4)	G(BARIS-TengahBaris-14,KOLOM+5) ...
G(BARIS-TengahBaris-14,KOLOM+6)	G(BARIS-TengahBaris-14,KOLOM+7)	G(BARIS-TengahBaris-14,KOLOM+8)...
G(BARIS-TengahBaris-14,KOLOM+9)	G(BARIS-TengahBaris-14,KOLOM+10)	G(BARIS-TengahBaris-14,KOLOM+11)...
G(BARIS-TengahBaris-14,KOLOM+12)	G(BARIS-TengahBaris-14,KOLOM+13)	G(BARIS-TengahBaris-14,KOLOM+14)...
G(BARIS-TengahBaris-14,KOLOM+15)	G(BARIS-TengahBaris-14,KOLOM+16)	G(BARIS-TengahBaris-14,KOLOM+17)...
G(BARIS-TengahBaris-14,KOLOM+18)	G(BARIS-TengahBaris-14,KOLOM+19)	G(BARIS-TengahBaris-14,KOLOM+20)...
G(BARIS-TengahBaris-14,KOLOM+21)	G(BARIS-TengahBaris-14,KOLOM+22);	
G(BARIS-TengahBaris-13,KOLOM-21)...	G(BARIS-TengahBaris-13,KOLOM-20)	G(BARIS-TengahBaris-13,KOLOM-19)
G(BARIS-TengahBaris-13,KOLOM-19)	G(BARIS-TengahBaris-13,KOLOM-17)	G(BARIS-TengahBaris-13,KOLOM-16)
G(BARIS-TengahBaris-13,KOLOM-16)	G(BARIS-TengahBaris-13,KOLOM-14)	G(BARIS-TengahBaris-13,KOLOM-13)
G(BARIS-TengahBaris-13,KOLOM-13)	G(BARIS-TengahBaris-13,KOLOM-11)	G(BARIS-TengahBaris-13,KOLOM-10)
G(BARIS-TengahBaris-13,KOLOM-10)	G(BARIS-TengahBaris-13,KOLOM-8)	G(BARIS-TengahBaris-13,KOLOM-7)
G(BARIS-TengahBaris-13,KOLOM-8)	G(BARIS-TengahBaris-13,KOLOM-4)...	G(BARIS-TengahBaris-13,KOLOM-3)
G(BARIS-TengahBaris-13,KOLOM-3)		

13,KOLOM-2) G(BARIS-TengahBaris-13,KOLOM-1) ...	
G(BARIS-TengahBaris-13,KOLOM-0)	G(BARIS-TengahBaris-13,KOLOM+1) G(BARIS-TengahBaris-13,KOLOM+2) ...
G(BARIS-TengahBaris-13,KOLOM+2)	G(BARIS-TengahBaris-13,KOLOM+3) G(BARIS-TengahBaris-13,KOLOM+4)
G(BARIS-TengahBaris-13,KOLOM+4)	G(BARIS-TengahBaris-13,KOLOM+5) ...
G(BARIS-TengahBaris-13,KOLOM+6)	G(BARIS-TengahBaris-13,KOLOM+7) G(BARIS-TengahBaris-13,KOLOM+8)...
G(BARIS-TengahBaris-13,KOLOM+9)	G(BARIS-TengahBaris-13,KOLOM+10) G(BARIS-TengahBaris-13,KOLOM+11)...
G(BARIS-TengahBaris-13,KOLOM+12)	G(BARIS-TengahBaris-13,KOLOM+13) G(BARIS-TengahBaris-13,KOLOM+14)...
G(BARIS-TengahBaris-13,KOLOM+15)	G(BARIS-TengahBaris-13,KOLOM+16) G(BARIS-TengahBaris-13,KOLOM+17)...
G(BARIS-TengahBaris-13,KOLOM+18)	G(BARIS-TengahBaris-13,KOLOM+19) G(BARIS-TengahBaris-13,KOLOM+20)...
G(BARIS-TengahBaris-13,KOLOM+21)	G(BARIS-TengahBaris-13,KOLOM+22);
G(BARIS-TengahBaris-12,KOLOM-21)...	
G(BARIS-TengahBaris-12,KOLOM-20)	G(BARIS-TengahBaris-12,KOLOM-19) G(BARIS-TengahBaris-12,KOLOM-18)...
G(BARIS-TengahBaris-12,KOLOM-17)	G(BARIS-TengahBaris-12,KOLOM-16) G(BARIS-TengahBaris-12,KOLOM-15)...
G(BARIS-TengahBaris-12,KOLOM-14)	G(BARIS-TengahBaris-12,KOLOM-13) G(BARIS-TengahBaris-12,KOLOM-12)...
G(BARIS-TengahBaris-12,KOLOM-11)	G(BARIS-TengahBaris-12,KOLOM-10) G(BARIS-TengahBaris-12,KOLOM-9)...
G(BARIS-TengahBaris-12,KOLOM-8)	G(BARIS-TengahBaris-12,KOLOM-7) G(BARIS-TengahBaris-12,KOLOM-6)...
G(BARIS-TengahBaris-12,KOLOM-4)...	
G(BARIS-TengahBaris-12,KOLOM-3)	G(BARIS-TengahBaris-12,KOLOM-2) G(BARIS-TengahBaris-12,KOLOM-1) ...
G(BARIS-TengahBaris-12,KOLOM-0)	G(BARIS-TengahBaris-12,KOLOM+1) G(BARIS-TengahBaris-12,KOLOM+2) ...
G(BARIS-TengahBaris-12,KOLOM+2)	G(BARIS-TengahBaris-12,KOLOM+3) G(BARIS-TengahBaris-12,KOLOM+4)
G(BARIS-TengahBaris-12,KOLOM+4)	G(BARIS-TengahBaris-12,KOLOM+5) ...
G(BARIS-TengahBaris-12,KOLOM+6)	G(BARIS-TengahBaris-12,KOLOM+7) G(BARIS-TengahBaris-12,KOLOM+8)...
G(BARIS-TengahBaris-12,KOLOM+9)	G(BARIS-TengahBaris-12,KOLOM+10) G(BARIS-TengahBaris-12,KOLOM+11)...

12,KOLOM+10) G(BARIS-TengahBaris-12,KOLOM+11)...  
     G(BARIS-TengahBaris-12,KOLOM+12)                      G(BARIS-TengahBaris-  
 12,KOLOM+13) G(BARIS-TengahBaris-12,KOLOM+14)...  
     G(BARIS-TengahBaris-12,KOLOM+15)                      G(BARIS-TengahBaris-  
 12,KOLOM+16) G(BARIS-TengahBaris-12,KOLOM+17)...  
     G(BARIS-TengahBaris-12,KOLOM+18)                      G(BARIS-TengahBaris-  
 12,KOLOM+19) G(BARIS-TengahBaris-12,KOLOM+20)...  
     G(BARIS-TengahBaris-12,KOLOM+21)                      G(BARIS-TengahBaris-  
 12,KOLOM+22);  
  
     G(BARIS-TengahBaris-11,KOLOM-21)...  
     G(BARIS-TengahBaris-11,KOLOM-20)                      G(BARIS-TengahBaris-  
 11,KOLOM-19) G(BARIS-TengahBaris-11,KOLOM-18)...  
     G(BARIS-TengahBaris-11,KOLOM-17)                      G(BARIS-TengahBaris-  
 11,KOLOM-16) G(BARIS-TengahBaris-11,KOLOM-15)...  
     G(BARIS-TengahBaris-11,KOLOM-14)                      G(BARIS-TengahBaris-  
 11,KOLOM-13) G(BARIS-TengahBaris-11,KOLOM-12)...  
     G(BARIS-TengahBaris-11,KOLOM-11)                      G(BARIS-TengahBaris-  
 11,KOLOM-10) G(BARIS-TengahBaris-11,KOLOM-9)...  
     G(BARIS-TengahBaris-11,KOLOM-8)                      G(BARIS-TengahBaris-  
 11,KOLOM-7) G(BARIS-TengahBaris-11,KOLOM-6)...  
     G(BARIS-TengahBaris-11,KOLOM-4)...  
     G(BARIS-TengahBaris-11,KOLOM-3)                      G(BARIS-TengahBaris-  
 11,KOLOM-2) G(BARIS-TengahBaris-11,KOLOM-1) ...  
     G(BARIS-TengahBaris-11,KOLOM-0)                      G(BARIS-TengahBaris-  
 11,KOLOM+1) G(BARIS-TengahBaris-11,KOLOM+2) ...  
     G(BARIS-TengahBaris-11,KOLOM+2)                      G(BARIS-TengahBaris-  
 11,KOLOM+3) G(BARIS-TengahBaris-11,KOLOM+3) ...  
     G(BARIS-TengahBaris-11,KOLOM+4)                      G(BARIS-TengahBaris-  
 11,KOLOM+4) G(BARIS-TengahBaris-11,KOLOM+5) ...  
     G(BARIS-TengahBaris-11,KOLOM+6)                      G(BARIS-TengahBaris-  
 11,KOLOM+7) G(BARIS-TengahBaris-11,KOLOM+8)...  
     G(BARIS-TengahBaris-11,KOLOM+9)                      G(BARIS-TengahBaris-  
 11,KOLOM+10) G(BARIS-TengahBaris-11,KOLOM+11)...  
     G(BARIS-TengahBaris-11,KOLOM+12)                      G(BARIS-TengahBaris-  
 11,KOLOM+13) G(BARIS-TengahBaris-11,KOLOM+14)...  
     G(BARIS-TengahBaris-11,KOLOM+15)                      G(BARIS-TengahBaris-  
 11,KOLOM+16) G(BARIS-TengahBaris-11,KOLOM+17)...  
     G(BARIS-TengahBaris-11,KOLOM+18)                      G(BARIS-TengahBaris-  
 11,KOLOM+19) G(BARIS-TengahBaris-11,KOLOM+20)...  
     G(BARIS-TengahBaris-11,KOLOM+21)                      G(BARIS-TengahBaris-  
 11,KOLOM+22);

G(BARIS-TengahBaris-10,KOLOM-21)...	
G(BARIS-TengahBaris-10,KOLOM-20)	G(BARIS-TengahBaris-10,KOLOM-19) G(BARIS-TengahBaris-10,KOLOM-18)...
G(BARIS-TengahBaris-10,KOLOM-17)	G(BARIS-TengahBaris-10,KOLOM-16) G(BARIS-TengahBaris-10,KOLOM-15)...
G(BARIS-TengahBaris-10,KOLOM-14)	G(BARIS-TengahBaris-10,KOLOM-13) G(BARIS-TengahBaris-10,KOLOM-12)...
G(BARIS-TengahBaris-10,KOLOM-11)	G(BARIS-TengahBaris-10,KOLOM-10) G(BARIS-TengahBaris-10,KOLOM-9)...
G(BARIS-TengahBaris-10,KOLOM-8)	G(BARIS-TengahBaris-10,KOLOM-7) G(BARIS-TengahBaris-10,KOLOM-6)...
G(BARIS-TengahBaris-10,KOLOM-4)...	
G(BARIS-TengahBaris-10,KOLOM-3)	G(BARIS-TengahBaris-10,KOLOM-2) G(BARIS-TengahBaris-10,KOLOM-1) ...
G(BARIS-TengahBaris-10,KOLOM-0)	G(BARIS-TengahBaris-10,KOLOM+1) G(BARIS-TengahBaris-10,KOLOM+2) ...
G(BARIS-TengahBaris-10,KOLOM+2)	G(BARIS-TengahBaris-10,KOLOM+3) G(BARIS-TengahBaris-10,KOLOM+3) ...
G(BARIS-TengahBaris-10,KOLOM+4)	G(BARIS-TengahBaris-10,KOLOM+4) G(BARIS--TengahBaris-10,KOLOM+5)...
G(BARIS-TengahBaris-10,KOLOM+6)	G(BARIS-TengahBaris-10,KOLOM+7) G(BARIS-TengahBaris-10,KOLOM+8)...
G(BARIS-TengahBaris-10,KOLOM+9)	G(BARIS-TengahBaris-10,KOLOM+10) G(BARIS-TengahBaris-10,KOLOM+11)...
G(BARIS-TengahBaris-10,KOLOM+12)	G(BARIS-TengahBaris-10,KOLOM+13) G(BARIS-TengahBaris-10,KOLOM+14)...
G(BARIS-TengahBaris-10,KOLOM+15)	G(BARIS-TengahBaris-10,KOLOM+16) G(BARIS-TengahBaris-10,KOLOM+17)...
G(BARIS-TengahBaris-10,KOLOM+18)	G(BARIS-TengahBaris-10,KOLOM+19) G(BARIS-TengahBaris-10,KOLOM+20)...
G(BARIS-TengahBaris-10,KOLOM+21)	G(BARIS-TengahBaris-10,KOLOM+22);
G(BARIS-TengahBaris-9,KOLOM-21)...	
G(BARIS-TengahBaris-9,KOLOM-20)	G(BARIS-TengahBaris-9,KOLOM-19) G(BARIS-TengahBaris-9,KOLOM-18)...
G(BARIS-TengahBaris-9,KOLOM-17)	G(BARIS-TengahBaris-9,KOLOM-16) G(BARIS-TengahBaris-9,KOLOM-15)...
G(BARIS-TengahBaris-9,KOLOM-14)	G(BARIS-TengahBaris-9,KOLOM-13) G(BARIS-TengahBaris-9,KOLOM-12)...
G(BARIS-TengahBaris-9,KOLOM-11)	G(BARIS-TengahBaris-9,KOLOM-10) G(BARIS-TengahBaris-9,KOLOM-9)...
G(BARIS-TengahBaris-9,KOLOM-8)	G(BARIS-TengahBaris-9,KOLOM-7) G(BARIS-TengahBaris-9,KOLOM-6)...



9,KOLOM-7) G(BARIS-TengahBaris-9,KOLOM-6)...	
G(BARIS-TengahBaris-9,KOLOM-4)...	
G(BARIS-TengahBaris-9,KOLOM-3)	G(BARIS-TengahBaris-
9,KOLOM-2) G(BARIS-TengahBaris-9,KOLOM-1) ...	
G(BARIS-TengahBaris-9,KOLOM-0)	G(BARIS-TengahBaris-
9,KOLOM+1) G(BARIS-TengahBaris-9,KOLOM+2) ...	
G(BARIS-TengahBaris-9,KOLOM+2)	G(BARIS-TengahBaris-
9,KOLOM+3) G(BARIS-TengahBaris-9,KOLOM+3) ...	
G(BARIS-TengahBaris-9,KOLOM+4)	G(BARIS-TengahBaris-
9,KOLOM+4) G(BARIS-TengahBaris-9,KOLOM+5) ...	
G(BARIS-TengahBaris-9,KOLOM+6)	G(BARIS-TengahBaris-
9,KOLOM+7) G(BARIS-TengahBaris-9,KOLOM+8) ...	
G(BARIS-TengahBaris-9,KOLOM+9)	G(BARIS-TengahBaris-
9,KOLOM+10) G(BARIS-TengahBaris-9,KOLOM+11)...	
G(BARIS-TengahBaris-9,KOLOM+12)	G(BARIS-TengahBaris-
9,KOLOM+13) G(BARIS-TengahBaris-9,KOLOM+14)...	
G(BARIS-TengahBaris-9,KOLOM+15)	G(BARIS-TengahBaris-
9,KOLOM+16) G(BARIS-TengahBaris-9,KOLOM+17)...	
G(BARIS-TengahBaris-9,KOLOM+18)	G(BARIS-TengahBaris-
9,KOLOM+19) G(BARIS-TengahBaris-9,KOLOM+20)...	
G(BARIS-TengahBaris-9,KOLOM+21)	G(BARIS-TengahBaris-
9,KOLOM+22);	
G(BARIS-TengahBaris-8,KOLOM-21)...	
G(BARIS-TengahBaris-8,KOLOM-20)	G(BARIS-TengahBaris-
8,KOLOM-19) G(BARIS-TengahBaris-8,KOLOM-18)...	
G(BARIS-TengahBaris-8,KOLOM-17)	G(BARIS-TengahBaris-
8,KOLOM-16) G(BARIS-TengahBaris-8,KOLOM-15)...	
G(BARIS-TengahBaris-8,KOLOM-14)	G(BARIS-TengahBaris-
8,KOLOM-13) G(BARIS-TengahBaris-8,KOLOM-12)...	
G(BARIS-TengahBaris-8,KOLOM-11)	G(BARIS-TengahBaris-
8,KOLOM-10) G(BARIS-TengahBaris-8,KOLOM-9)...	
G(BARIS-TengahBaris-8,KOLOM-8)	G(BARIS-TengahBaris-
8,KOLOM-7) G(BARIS-TengahBaris-8,KOLOM-6)...	
G(BARIS-TengahBaris-8,KOLOM-4)...	
G(BARIS-TengahBaris-8,KOLOM-3)	G(BARIS-TengahBaris-
8,KOLOM-2) G(BARIS-TengahBaris-8,KOLOM-1) ...	
G(BARIS-TengahBaris-8,KOLOM-0)	G(BARIS-TengahBaris-
8,KOLOM+1) G(BARIS-TengahBaris-8,KOLOM+2) ...	
G(BARIS-TengahBaris-8,KOLOM+2)	G(BARIS-TengahBaris-
8,KOLOM+3) G(BARIS-TengahBaris-8,KOLOM+3) ...	
G(BARIS-TengahBaris-8,KOLOM+4)	G(BARIS-TengahBaris-
8,KOLOM+4) G(BARIS-TengahBaris-8,KOLOM+5) ...	

G(BARIS-TengahBaris-8,KOLOM+6)	G(BARIS-TengahBaris-8,KOLOM+7) G(BARIS-TengahBaris-8,KOLOM+8)...	G(BARIS-TengahBaris-8,KOLOM+9)	G(BARIS-TengahBaris-8,KOLOM+10) G(BARIS-TengahBaris-8,KOLOM+11)...
G(BARIS-TengahBaris-8,KOLOM+12)	G(BARIS-TengahBaris-8,KOLOM+13) G(BARIS-TengahBaris-8,KOLOM+14)...	G(BARIS-TengahBaris-8,KOLOM+15)	G(BARIS-TengahBaris-8,KOLOM+16) G(BARIS-TengahBaris-8,KOLOM+17)...
G(BARIS-TengahBaris-8,KOLOM+18)	G(BARIS-TengahBaris-8,KOLOM+19) G(BARIS-TengahBaris-8,KOLOM+20)...	G(BARIS-TengahBaris-8,KOLOM+21)	G(BARIS-TengahBaris-8,KOLOM+22);
G(BARIS-TengahBaris-7,KOLOM-21)...	G(BARIS-TengahBaris-7,KOLOM-20)	G(BARIS-TengahBaris-7,KOLOM-19) G(BARIS-TengahBaris-7,KOLOM-18)...	G(BARIS-TengahBaris-7,KOLOM-17)
G(BARIS-TengahBaris-7,KOLOM-16) G(BARIS-TengahBaris-7,KOLOM-15)...	G(BARIS-TengahBaris-7,KOLOM-14)	G(BARIS-TengahBaris-7,KOLOM-13) G(BARIS-TengahBaris-7,KOLOM-12)...	G(BARIS-TengahBaris-7,KOLOM-11)
G(BARIS-TengahBaris-7,KOLOM-10) G(BARIS-TengahBaris-7,KOLOM-9)...	G(BARIS-TengahBaris-7,KOLOM-8)	G(BARIS-TengahBaris-7,KOLOM-7) G(BARIS-TengahBaris-7,KOLOM-6)...	G(BARIS-TengahBaris-7,KOLOM-4)...
G(BARIS-TengahBaris-7,KOLOM-3)	G(BARIS-TengahBaris-7,KOLOM-2) G(BARIS-TengahBaris-7,KOLOM-1) ...	G(BARIS-TengahBaris-7,KOLOM+1) G(BARIS-TengahBaris-7,KOLOM+2) ...	G(BARIS-TengahBaris-7,KOLOM+3) ...
G(BARIS-TengahBaris-7,KOLOM+4)	G(BARIS-TengahBaris-7,KOLOM+5) ...	G(BARIS-TengahBaris-7,KOLOM+6)	G(BARIS-TengahBaris-7,KOLOM+7) G(BARIS-TengahBaris-7,KOLOM+8)...
G(BARIS-TengahBaris-7,KOLOM+10) G(BARIS-TengahBaris-7,KOLOM+11)...	G(BARIS-TengahBaris-7,KOLOM+12)	G(BARIS-TengahBaris-7,KOLOM+13) G(BARIS-TengahBaris-7,KOLOM+14)...	G(BARIS-TengahBaris-7,KOLOM+15)
G(BARIS-TengahBaris-7,KOLOM+16) G(BARIS-TengahBaris-7,KOLOM+17)...	G(BARIS-TengahBaris-7,KOLOM+18)	G(BARIS-TengahBaris-7,KOLOM+19) G(BARIS-TengahBaris-7,KOLOM+20)...	

G(BARIS-TengahBaris-7,KOLOM+21) 7,KOLOM+22);	G(BARIS-TengahBaris-
G(BARIS-TengahBaris-6,KOLOM-21)...	
G(BARIS-TengahBaris-6,KOLOM-20)	G(BARIS-TengahBaris-
6,KOLOM-19) G(BARIS-TengahBaris-6,KOLOM-18)...	
G(BARIS-TengahBaris-6,KOLOM-17)	G(BARIS-TengahBaris-
6,KOLOM-16) G(BARIS-TengahBaris-6,KOLOM-15)...	
G(BARIS-TengahBaris-6,KOLOM-14)	G(BARIS-TengahBaris-
6,KOLOM-13) G(BARIS-TengahBaris-6,KOLOM-12)...	
G(BARIS-TengahBaris-6,KOLOM-11)	G(BARIS-TengahBaris-
6,KOLOM-10) G(BARIS-TengahBaris-6,KOLOM-9)...	
G(BARIS-TengahBaris-6,KOLOM-8)	G(BARIS-TengahBaris-
6,KOLOM-7) G(BARIS-TengahBaris-6,KOLOM-6)...	
G(BARIS-TengahBaris-6,KOLOM-4)...	
G(BARIS-TengahBaris-6,KOLOM-3)	G(BARIS-TengahBaris-
6,KOLOM-2) G(BARIS-TengahBaris-6,KOLOM-1) ...	
G(BARIS-TengahBaris-6,KOLOM-0)	G(BARIS-TengahBaris-
6,KOLOM+1) G(BARIS-TengahBaris-6,KOLOM+2) ...	
G(BARIS-TengahBaris-6,KOLOM+2)	G(BARIS-TengahBaris-
6,KOLOM+3) G(BARIS-TengahBaris-6,KOLOM+3) ...	
G(BARIS-TengahBaris-6,KOLOM+4)	G(BARIS-TengahBaris-
6,KOLOM+4) G(BARIS-TengahBaris-6,KOLOM+5) ...	
G(BARIS-TengahBaris-6,KOLOM+6)	G(BARIS-TengahBaris-
6,KOLOM+7) G(BARIS-TengahBaris-6,KOLOM+8)...	
G(BARIS-TengahBaris-6,KOLOM+9)	G(BARIS-TengahBaris-
6,KOLOM+10) G(BARIS-TengahBaris-6,KOLOM+11)...	
G(BARIS-TengahBaris-6,KOLOM+12)	G(BARIS-TengahBaris-
6,KOLOM+13) G(BARIS-TengahBaris-6,KOLOM+14)...	
G(BARIS-TengahBaris-6,KOLOM+15)	G(BARIS-TengahBaris-
6,KOLOM+16) G(BARIS-TengahBaris-6,KOLOM+17)...	
G(BARIS-TengahBaris-6,KOLOM+18)	G(BARIS-TengahBaris-
6,KOLOM+19) G(BARIS-TengahBaris-6,KOLOM+20)...	
G(BARIS-TengahBaris-6,KOLOM+21)	G(BARIS-TengahBaris-
6,KOLOM+22);	
G(BARIS-TengahBaris-5,KOLOM-21)...	
G(BARIS-TengahBaris-5,KOLOM-20)	G(BARIS-TengahBaris-
5,KOLOM-19) G(BARIS-TengahBaris-5,KOLOM-18)...	
G(BARIS-TengahBaris-5,KOLOM-17)	G(BARIS-TengahBaris-
5,KOLOM-16) G(BARIS-TengahBaris-5,KOLOM-15)...	
G(BARIS-TengahBaris-5,KOLOM-14)	G(BARIS-TengahBaris-
5,KOLOM-13) G(BARIS-TengahBaris-5,KOLOM-12)...	

G(BARIS-TengahBaris-5,KOLOM-11)	G(BARIS-TengahBaris-5,KOLOM-10) G(BARIS-TengahBaris-5,KOLOM-9)...
G(BARIS-TengahBaris-5,KOLOM-8)	G(BARIS-TengahBaris-5,KOLOM-7) G(BARIS-TengahBaris-5,KOLOM-6)...
G(BARIS-TengahBaris-5,KOLOM-4)...	G(BARIS-TengahBaris-5,KOLOM-3)
G(BARIS-TengahBaris-5,KOLOM-2)	G(BARIS-TengahBaris-5,KOLOM-1) ...
G(BARIS-TengahBaris-5,KOLOM-0)	G(BARIS-TengahBaris-5,KOLOM+1) G(BARIS-TengahBaris-5,KOLOM+2) ...
G(BARIS-TengahBaris-5,KOLOM+2)	G(BARIS-TengahBaris-5,KOLOM+3) ...
G(BARIS-TengahBaris-5,KOLOM+4)	G(BARIS-TengahBaris-5,KOLOM+5) ...
G(BARIS-TengahBaris-5,KOLOM+6)	G(BARIS-TengahBaris-5,KOLOM+7) G(BARIS-TengahBaris-5,KOLOM+8)...
G(BARIS-TengahBaris-5,KOLOM+9)	G(BARIS-TengahBaris-5,KOLOM+10) G(BARIS-TengahBaris-5,KOLOM+11)...
G(BARIS-TengahBaris-5,KOLOM+12)	G(BARIS-TengahBaris-5,KOLOM+13) G(BARIS-TengahBaris-5,KOLOM+14)...
G(BARIS-TengahBaris-5,KOLOM+15)	G(BARIS-TengahBaris-5,KOLOM+16) G(BARIS-TengahBaris-5,KOLOM+17)...
G(BARIS-TengahBaris-5,KOLOM+18)	G(BARIS-TengahBaris-5,KOLOM+19) G(BARIS-TengahBaris-5,KOLOM+20)...
G(BARIS-TengahBaris-5,KOLOM+21)	G(BARIS-TengahBaris-5,KOLOM+22);
G(BARIS-TengahBaris-4,KOLOM-21)...	G(BARIS-TengahBaris-4,KOLOM-20)
G(BARIS-TengahBaris-4,KOLOM-19)	G(BARIS-TengahBaris-4,KOLOM-18)...
G(BARIS-TengahBaris-4,KOLOM-17)	G(BARIS-TengahBaris-4,KOLOM-16) G(BARIS-TengahBaris-4,KOLOM-15)...
G(BARIS-TengahBaris-4,KOLOM-14)	G(BARIS-TengahBaris-4,KOLOM-13) G(BARIS-TengahBaris-4,KOLOM-12)...
G(BARIS-TengahBaris-4,KOLOM-11)	G(BARIS-TengahBaris-4,KOLOM-10) G(BARIS-TengahBaris-4,KOLOM-9)...
G(BARIS-TengahBaris-4,KOLOM-8)	G(BARIS-TengahBaris-4,KOLOM-7) G(BARIS-TengahBaris-4,KOLOM-6)...
G(BARIS-TengahBaris-4,KOLOM-4)...	G(BARIS-TengahBaris-4,KOLOM-3)
G(BARIS-TengahBaris-4,KOLOM-2)	G(BARIS-TengahBaris-4,KOLOM-1) ...
G(BARIS-TengahBaris-4,KOLOM-0)	G(BARIS-TengahBaris-4,KOLOM+1) G(BARIS-TengahBaris-4,KOLOM+2) ...
G(BARIS-TengahBaris-4,KOLOM+2)	

4,KOLOM+3) G(BARIS-TengahBaris-4,KOLOM+3) ...	
G(BARIS-TengahBaris-4,KOLOM+4)	G(BARIS-TengahBaris-4,KOLOM+4) G(BARIS-TengahBaris-4,KOLOM+5) ...
G(BARIS-TengahBaris-4,KOLOM+6)	G(BARIS-TengahBaris-4,KOLOM+7) G(BARIS-TengahBaris-4,KOLOM+8)...
G(BARIS-TengahBaris-4,KOLOM+9)	G(BARIS-TengahBaris-4,KOLOM+10) G(BARIS-TengahBaris-4,KOLOM+11)...
G(BARIS-TengahBaris-4,KOLOM+12)	G(BARIS-TengahBaris-4,KOLOM+13) G(BARIS-TengahBaris-4,KOLOM+14)...
G(BARIS-TengahBaris-4,KOLOM+15)	G(BARIS-TengahBaris-4,KOLOM+16) G(BARIS-TengahBaris-4,KOLOM+17)...
G(BARIS-TengahBaris-4,KOLOM+18)	G(BARIS-TengahBaris-4,KOLOM+19) G(BARIS-TengahBaris-4,KOLOM+20)...
G(BARIS-TengahBaris-4,KOLOM+21)	G(BARIS-TengahBaris-4,KOLOM+22);
G(BARIS-TengahBaris-3,KOLOM-21)...	
G(BARIS-TengahBaris-3,KOLOM-20)	G(BARIS-TengahBaris-3,KOLOM-19) G(BARIS-TengahBaris-3,KOLOM-18)...
G(BARIS-TengahBaris-3,KOLOM-17)	G(BARIS-TengahBaris-3,KOLOM-16) G(BARIS-TengahBaris-3,KOLOM-15)...
G(BARIS-TengahBaris-3,KOLOM-14)	G(BARIS-TengahBaris-3,KOLOM-13) G(BARIS-TengahBaris-3,KOLOM-12)...
G(BARIS-TengahBaris-3,KOLOM-11)	G(BARIS-TengahBaris-3,KOLOM-10) G(BARIS-TengahBaris-3,KOLOM-9)...
G(BARIS-TengahBaris-3,KOLOM-8)	G(BARIS-TengahBaris-3,KOLOM-7) G(BARIS-TengahBaris-3,KOLOM-6)...
G(BARIS-TengahBaris-3,KOLOM-4)...	
G(BARIS-TengahBaris-3,KOLOM-3)	G(BARIS-TengahBaris-3,KOLOM-2) G(BARIS-TengahBaris-3,KOLOM-1) ...
G(BARIS-TengahBaris-3,KOLOM-0)	G(BARIS-TengahBaris-3,KOLOM+1) G(BARIS-TengahBaris-3,KOLOM+2) ...
G(BARIS-TengahBaris-3,KOLOM+2)	G(BARIS-TengahBaris-3,KOLOM+3) G(BARIS-TengahBaris-3,KOLOM+4)
G(BARIS-TengahBaris-3,KOLOM+5)	G(BARIS-TengahBaris-3,KOLOM+6)
G(BARIS-TengahBaris-3,KOLOM+7)	G(BARIS-TengahBaris-3,KOLOM+8)...
G(BARIS-TengahBaris-3,KOLOM+9)	G(BARIS-TengahBaris-3,KOLOM+10) G(BARIS-TengahBaris-3,KOLOM+11)...
G(BARIS-TengahBaris-3,KOLOM+12)	G(BARIS-TengahBaris-3,KOLOM+13) G(BARIS-TengahBaris-3,KOLOM+14)...
G(BARIS-TengahBaris-3,KOLOM+15)	G(BARIS-TengahBaris-3,KOLOM+15)

3,KOLOM+16) G(BARIS-TengahBaris-3,KOLOM+17)...	
G(BARIS-TengahBaris-3,KOLOM+18)	G(BARIS-TengahBaris-3,KOLOM+19) G(BARIS-TengahBaris-3,KOLOM+20)...
G(BARIS-TengahBaris-3,KOLOM+21)	G(BARIS-TengahBaris-3,KOLOM+22);
G(BARIS-TengahBaris-2,KOLOM-21)...	
G(BARIS-TengahBaris-2,KOLOM-20)	G(BARIS-TengahBaris-2,KOLOM-19) G(BARIS-TengahBaris-2,KOLOM-18)...
G(BARIS-TengahBaris-2,KOLOM-17)	G(BARIS-TengahBaris-2,KOLOM-16) G(BARIS-TengahBaris-2,KOLOM-15)...
G(BARIS-TengahBaris-2,KOLOM-14)	G(BARIS-TengahBaris-2,KOLOM-13) G(BARIS-TengahBaris-2,KOLOM-12)...
G(BARIS-TengahBaris-2,KOLOM-11)	G(BARIS-TengahBaris-2,KOLOM-10) G(BARIS-TengahBaris-2,KOLOM-9)...
G(BARIS-TengahBaris-2,KOLOM-8)	G(BARIS-TengahBaris-2,KOLOM-7) G(BARIS-TengahBaris-2,KOLOM-6)...
G(BARIS-TengahBaris-2,KOLOM-4)...	
G(BARIS-TengahBaris-2,KOLOM-3)	G(BARIS-TengahBaris-2,KOLOM-2) G(BARIS-TengahBaris-2,KOLOM-1) ...
G(BARIS-TengahBaris-2,KOLOM-0)	G(BARIS-TengahBaris-2,KOLOM+1) G(BARIS-TengahBaris-2,KOLOM+2) ...
G(BARIS-TengahBaris-2,KOLOM+2)	G(BARIS-TengahBaris-2,KOLOM+3) G(BARIS-TengahBaris-2,KOLOM+3) ...
G(BARIS-TengahBaris-2,KOLOM+4)	G(BARIS-TengahBaris-2,KOLOM+4) G(BARIS-TengahBaris-2,KOLOM+5) ...
G(BARIS-TengahBaris-2,KOLOM+6)	G(BARIS-TengahBaris-2,KOLOM+7) G(BARIS-TengahBaris-2,KOLOM+8)...
G(BARIS-TengahBaris-2,KOLOM+9)	G(BARIS-TengahBaris-2,KOLOM+10) G(BARIS-TengahBaris-2,KOLOM+11)...
G(BARIS-TengahBaris-2,KOLOM+12)	G(BARIS-TengahBaris-2,KOLOM+13) G(BARIS-TengahBaris-2,KOLOM+14)...
G(BARIS-TengahBaris-2,KOLOM+15)	G(BARIS-TengahBaris-2,KOLOM+16) G(BARIS-TengahBaris-2,KOLOM+17)...
G(BARIS-TengahBaris-2,KOLOM+18)	G(BARIS-TengahBaris-2,KOLOM+19) G(BARIS-TengahBaris-2,KOLOM+20)...
G(BARIS-TengahBaris-2,KOLOM+21)	G(BARIS-TengahBaris-2,KOLOM+22);
G(BARIS-TengahBaris-1,KOLOM-21)...	
G(BARIS-TengahBaris-1,KOLOM-20)	G(BARIS-TengahBaris-1,KOLOM-19) G(BARIS-TengahBaris-1,KOLOM-18)...
G(BARIS-TengahBaris-1,KOLOM-17)	G(BARIS-TengahBaris-1,KOLOM-16) G(BARIS-TengahBaris-1,KOLOM-15)...
G(BARIS-TengahBaris-1,KOLOM-14)	G(BARIS-TengahBaris-1,KOLOM-13) G(BARIS-TengahBaris-1,KOLOM-12)...
G(BARIS-TengahBaris-1,KOLOM-11)	G(BARIS-TengahBaris-1,KOLOM-10) G(BARIS-TengahBaris-1,KOLOM-9)...
G(BARIS-TengahBaris-1,KOLOM-8)	G(BARIS-TengahBaris-1,KOLOM-7) G(BARIS-TengahBaris-1,KOLOM-6)...
G(BARIS-TengahBaris-1,KOLOM-4)...	
G(BARIS-TengahBaris-1,KOLOM-3)	G(BARIS-TengahBaris-1,KOLOM-2) G(BARIS-TengahBaris-1,KOLOM-1) ...
G(BARIS-TengahBaris-1,KOLOM-0)	G(BARIS-TengahBaris-1,KOLOM+1) G(BARIS-TengahBaris-1,KOLOM+2) ...
G(BARIS-TengahBaris-1,KOLOM+2)	G(BARIS-TengahBaris-1,KOLOM+3) G(BARIS-TengahBaris-1,KOLOM+3) ...
G(BARIS-TengahBaris-1,KOLOM+4)	G(BARIS-TengahBaris-1,KOLOM+4) G(BARIS-TengahBaris-1,KOLOM+5) ...
G(BARIS-TengahBaris-1,KOLOM+6)	G(BARIS-TengahBaris-1,KOLOM+7) G(BARIS-TengahBaris-1,KOLOM+8)...
G(BARIS-TengahBaris-1,KOLOM+9)	G(BARIS-TengahBaris-1,KOLOM+10) G(BARIS-TengahBaris-1,KOLOM+11)...
G(BARIS-TengahBaris-1,KOLOM+12)	G(BARIS-TengahBaris-1,KOLOM+13) G(BARIS-TengahBaris-1,KOLOM+14)...
G(BARIS-TengahBaris-1,KOLOM+15)	G(BARIS-TengahBaris-1,KOLOM+16) G(BARIS-TengahBaris-1,KOLOM+17)...
G(BARIS-TengahBaris-1,KOLOM+18)	G(BARIS-TengahBaris-1,KOLOM+19) G(BARIS-TengahBaris-1,KOLOM+20)...
G(BARIS-TengahBaris-1,KOLOM+21)	G(BARIS-TengahBaris-1,KOLOM+22);

1,KOLOM-16) G(BARIS-TengahBaris-1,KOLOM-15)...	
G(BARIS-TengahBaris-1,KOLOM-14)	G(BARIS-TengahBaris-1,KOLOM-13) G(BARIS-TengahBaris-1,KOLOM-12)...
G(BARIS-TengahBaris-1,KOLOM-11)	G(BARIS-TengahBaris-1,KOLOM-10) G(BARIS-TengahBaris-1,KOLOM-9)...
G(BARIS-TengahBaris-1,KOLOM-8)	G(BARIS-TengahBaris-1,KOLOM-7) G(BARIS-TengahBaris-1,KOLOM-6)...
G(BARIS-TengahBaris-1,KOLOM-4)...	G(BARIS-TengahBaris-1,KOLOM-3)
G(BARIS-TengahBaris-1,KOLOM-2)	G(BARIS-TengahBaris-1,KOLOM-1) ...
G(BARIS-TengahBaris-1,KOLOM-0)	G(BARIS-TengahBaris-1,KOLOM+1) G(BARIS-TengahBaris-1,KOLOM+2) ...
G(BARIS-TengahBaris-1,KOLOM+2)	G(BARIS-TengahBaris-1,KOLOM+3) G(BARIS-TengahBaris-1,KOLOM+3) ...
G(BARIS-TengahBaris-1,KOLOM+4)	G(BARIS-TengahBaris-1,KOLOM+4) G(BARIS-TengahBaris-1,KOLOM+5) ...
G(BARIS-TengahBaris-1,KOLOM+6)	G(BARIS-TengahBaris-1,KOLOM+7) G(BARIS-TengahBaris-1,KOLOM+8)...
G(BARIS-TengahBaris-1,KOLOM+9)	G(BARIS-TengahBaris-1,KOLOM+10) G(BARIS-TengahBaris-1,KOLOM+11)...
G(BARIS-TengahBaris-1,KOLOM+12)	G(BARIS-TengahBaris-1,KOLOM+13) G(BARIS-TengahBaris-1,KOLOM+14)...
G(BARIS-TengahBaris-1,KOLOM+15)	G(BARIS-TengahBaris-1,KOLOM+16) G(BARIS-TengahBaris-1,KOLOM+17)...
G(BARIS-TengahBaris-1,KOLOM+18)	G(BARIS-TengahBaris-1,KOLOM+19) G(BARIS-TengahBaris-1,KOLOM+20)...
G(BARIS-TengahBaris-1,KOLOM+21)	G(BARIS-TengahBaris-1,KOLOM+22);
G(BARIS-TengahBaris,KOLOM-21)...	
G(BARIS-TengahBaris,KOLOM-20)	G(BARIS-TengahBaris,KOLOM-19) G(BARIS-TengahBaris,KOLOM-18)...
G(BARIS-TengahBaris,KOLOM-17)	G(BARIS-TengahBaris,KOLOM-16) G(BARIS-TengahBaris,KOLOM-15)...
G(BARIS-TengahBaris,KOLOM-14)	G(BARIS-TengahBaris,KOLOM-13) G(BARIS-TengahBaris,KOLOM-12)...
G(BARIS-TengahBaris,KOLOM-11)	G(BARIS-TengahBaris,KOLOM-10) G(BARIS-TengahBaris,KOLOM-9)...
G(BARIS-TengahBaris,KOLOM-8)	G(BARIS-TengahBaris,KOLOM-7) G(BARIS-TengahBaris,KOLOM-6)...
G(BARIS-TengahBaris,KOLOM-4)...	G(BARIS-TengahBaris,KOLOM-3)
G(BARIS-TengahBaris,KOLOM-2)	G(BARIS-TengahBaris,KOLOM-1) ...

G(BARIS-TengahBaris,KOLOM-0)	G(BARIS-TengahBaris,KOLOM+1)
G(BARIS-TengahBaris,KOLOM+2) ...	
G(BARIS-TengahBaris,KOLOM+2)	G(BARIS-TengahBaris,KOLOM+3)
G(BARIS-TengahBaris,KOLOM+3) ...	
G(BARIS-TengahBaris,KOLOM+4)	G(BARIS-TengahBaris,KOLOM+4)
G(BARIS-TengahBaris,KOLOM+5) ...	
G(BARIS-TengahBaris,KOLOM+6)	G(BARIS-TengahBaris,KOLOM+7)
G(BARIS-TengahBaris,KOLOM+8)...	
G(BARIS-TengahBaris,KOLOM+9)	G(BARIS-TengahBaris,KOLOM+10)
G(BARIS-TengahBaris,KOLOM+10)	G(BARIS-TengahBaris,KOLOM+11)...
G(BARIS-TengahBaris,KOLOM+12)	G(BARIS-TengahBaris,KOLOM+13)
G(BARIS-TengahBaris,KOLOM+13)	G(BARIS-TengahBaris,KOLOM+14)...
G(BARIS-TengahBaris,KOLOM+15)	G(BARIS-TengahBaris,KOLOM+16)
G(BARIS-TengahBaris,KOLOM+16)	G(BARIS-TengahBaris,KOLOM+17)...
G(BARIS-TengahBaris,KOLOM+18)	G(BARIS-TengahBaris,KOLOM+19)
G(BARIS-TengahBaris,KOLOM+19)	G(BARIS-TengahBaris,KOLOM+20)...
G(BARIS-TengahBaris,KOLOM+21)	G(BARIS-TengahBaris,KOLOM+22);
G(BARIS-TengahBaris+1,KOLOM-21)...	
G(BARIS-TengahBaris+1,KOLOM-20)	G(BARIS-TengahBaris+1,KOLOM-19)
G(BARIS-TengahBaris+1,KOLOM-19)	G(BARIS-TengahBaris+1,KOLOM-18)...
G(BARIS-TengahBaris+1,KOLOM-17)	G(BARIS-TengahBaris+1,KOLOM-16)
G(BARIS-TengahBaris+1,KOLOM-16)	G(BARIS-TengahBaris+1,KOLOM-15)...
G(BARIS-TengahBaris+1,KOLOM-14)	G(BARIS-TengahBaris+1,KOLOM-13)
G(BARIS-TengahBaris+1,KOLOM-13)	G(BARIS-TengahBaris+1,KOLOM-12)...
G(BARIS-TengahBaris+1,KOLOM-11)	G(BARIS-TengahBaris+1,KOLOM-10)
G(BARIS-TengahBaris+1,KOLOM-10)	G(BARIS-TengahBaris+1,KOLOM-9)...
G(BARIS-TengahBaris+1,KOLOM-8)	G(BARIS-TengahBaris+1,KOLOM-7)
G(BARIS-TengahBaris+1,KOLOM-7)	G(BARIS-TengahBaris+1,KOLOM-6)...
G(BARIS-TengahBaris+1,KOLOM-4)...	
G(BARIS-TengahBaris+1,KOLOM-3)	G(BARIS-TengahBaris+1,KOLOM-2)
G(BARIS-TengahBaris+1,KOLOM-2)	G(BARIS-TengahBaris+1,KOLOM-1) ...
G(BARIS-TengahBaris+1,KOLOM-0)	G(BARIS-TengahBaris+1,KOLOM+1)
G(BARIS-TengahBaris+1,KOLOM+1)	G(BARIS-TengahBaris+1,KOLOM+2) ...
G(BARIS-TengahBaris+1,KOLOM+2)	G(BARIS-TengahBaris+1,KOLOM+3)
G(BARIS-TengahBaris+1,KOLOM+3)	G(BARIS-TengahBaris+1,KOLOM+4)
G(BARIS-TengahBaris+1,KOLOM+4)	G(BARIS-TengahBaris+1,KOLOM+5) ...
G(BARIS-TengahBaris+1,KOLOM+6)	G(BARIS-TengahBaris+1,KOLOM+7)
G(BARIS-TengahBaris+1,KOLOM+7)	G(BARIS-TengahBaris+1,KOLOM+8)...
G(BARIS-TengahBaris+1,KOLOM+9)	G(BARIS-TengahBaris+1,KOLOM+10)
G(BARIS-TengahBaris+1,KOLOM+10)	G(BARIS-TengahBaris+1,KOLOM+11)...





G(BARIS-TengahBaris+3,KOLOM-20)	G(BARIS-TengahBaris+3,KOLOM-19)	G(BARIS-TengahBaris+3,KOLOM-18)
G(BARIS-TengahBaris+3,KOLOM-17)	G(BARIS-TengahBaris+3,KOLOM-16)	G(BARIS-TengahBaris+3,KOLOM-15)...
G(BARIS-TengahBaris+3,KOLOM-14)	G(BARIS-TengahBaris+3,KOLOM-13)	G(BARIS-TengahBaris+3,KOLOM-12)...
G(BARIS-TengahBaris+3,KOLOM-11)	G(BARIS-TengahBaris+3,KOLOM-10)	G(BARIS-TengahBaris+3,KOLOM-9)...
G(BARIS-TengahBaris+3,KOLOM-8)	G(BARIS-TengahBaris+3,KOLOM-7)	G(BARIS-TengahBaris+3,KOLOM-6)...
G(BARIS-TengahBaris+3,KOLOM-4)...	G(BARIS-TengahBaris+3,KOLOM-3)	G(BARIS-TengahBaris+3,KOLOM-2)
G(BARIS-TengahBaris+3,KOLOM-2)	G(BARIS-TengahBaris+3,KOLOM-1)	G(BARIS-TengahBaris+3,KOLOM-0)
G(BARIS-TengahBaris+3,KOLOM+1)	G(BARIS-TengahBaris+3,KOLOM+2)	G(BARIS-TengahBaris+3,KOLOM+3)
G(BARIS-TengahBaris+3,KOLOM+2)	G(BARIS-TengahBaris+3,KOLOM+3)	G(BARIS-TengahBaris+3,KOLOM+4)
G(BARIS-TengahBaris+3,KOLOM+4)	G(BARIS-TengahBaris+3,KOLOM+5)	G(BARIS-TengahBaris+3,KOLOM+6)
G(BARIS-TengahBaris+3,KOLOM+7)	G(BARIS-TengahBaris+3,KOLOM+8)	G(BARIS-TengahBaris+3,KOLOM+9)
G(BARIS-TengahBaris+3,KOLOM+10)	G(BARIS-TengahBaris+3,KOLOM+11)	G(BARIS-TengahBaris+3,KOLOM+12)
G(BARIS-TengahBaris+3,KOLOM+13)	G(BARIS-TengahBaris+3,KOLOM+14)	G(BARIS-TengahBaris+3,KOLOM+15)
G(BARIS-TengahBaris+3,KOLOM+16)	G(BARIS-TengahBaris+3,KOLOM+17)	G(BARIS-TengahBaris+3,KOLOM+18)
G(BARIS-TengahBaris+3,KOLOM+19)	G(BARIS-TengahBaris+3,KOLOM+20)	G(BARIS-TengahBaris+3,KOLOM+21)
G(BARIS-TengahBaris+3,KOLOM+22);		
G(BARIS-TengahBaris+4,KOLOM-21)...	G(BARIS-TengahBaris+4,KOLOM-20)	G(BARIS-TengahBaris+4,KOLOM-19)
G(BARIS-TengahBaris+4,KOLOM-19)	G(BARIS-TengahBaris+4,KOLOM-18)	G(BARIS-TengahBaris+4,KOLOM-17)
G(BARIS-TengahBaris+4,KOLOM-16)	G(BARIS-TengahBaris+4,KOLOM-15)	G(BARIS-TengahBaris+4,KOLOM-14)
G(BARIS-TengahBaris+4,KOLOM-13)	G(BARIS-TengahBaris+4,KOLOM-12)	G(BARIS-TengahBaris+4,KOLOM-11)
G(BARIS-TengahBaris+4,KOLOM-10)	G(BARIS-TengahBaris+4,KOLOM-9)	G(BARIS-TengahBaris+4,KOLOM-8)
G(BARIS-TengahBaris+4,KOLOM-7)	G(BARIS-TengahBaris+4,KOLOM-6)	

G(BARIS-TengahBaris+4,KOLOM-4)...	
G(BARIS-TengahBaris+4,KOLOM-3)	G(BARIS-
TengahBaris+4,KOLOM-2) G(BARIS-TengahBaris+4,KOLOM-1) ...	
G(BARIS-TengahBaris+4,KOLOM-0)	G(BARIS-
TengahBaris+4,KOLOM+1) G(BARIS-TengahBaris+4,KOLOM+2) ...	
G(BARIS-TengahBaris+4,KOLOM+2)	G(BARIS-
TengahBaris+4,KOLOM+3) G(BARIS-TengahBaris+4,KOLOM+3) ...	
G(BARIS-TengahBaris+4,KOLOM+4)	G(BARIS-
TengahBaris+4,KOLOM+4) G(BARIS-TengahBaris+4,KOLOM+5) ...	
G(BARIS-TengahBaris+4,KOLOM+6)	G(BARIS-
TengahBaris+4,KOLOM+7) G(BARIS-TengahBaris+4,KOLOM+8)...	
G(BARIS-TengahBaris+4,KOLOM+9)	G(BARIS-
TengahBaris+4,KOLOM+10) G(BARIS-TengahBaris+4,KOLOM+11)...	
G(BARIS-TengahBaris+4,KOLOM+12)	G(BARIS-
TengahBaris+4,KOLOM+13) G(BARIS-TengahBaris+4,KOLOM+14)...	
G(BARIS-TengahBaris+4,KOLOM+15)	G(BARIS-
TengahBaris+4,KOLOM+16) G(BARIS-TengahBaris+4,KOLOM+17)...	
G(BARIS-TengahBaris+4,KOLOM+18)	G(BARIS-
TengahBaris+4,KOLOM+19) G(BARIS-TengahBaris+4,KOLOM+20)...	
G(BARIS-TengahBaris+4,KOLOM+21)	G(BARIS-
TengahBaris+4,KOLOM+22);	
G(BARIS-TengahBaris+5,KOLOM-21)...	
G(BARIS-TengahBaris+5,KOLOM-20)	G(BARIS-
TengahBaris+5,KOLOM-19) G(BARIS-TengahBaris+5,KOLOM-18)...	
G(BARIS-TengahBaris+5,KOLOM-17)	G(BARIS-
TengahBaris+5,KOLOM-16) G(BARIS-TengahBaris+5,KOLOM-15)...	
G(BARIS-TengahBaris+5,KOLOM-14)	G(BARIS-
TengahBaris+5,KOLOM-13) G(BARIS-TengahBaris+5,KOLOM-12)...	
G(BARIS-TengahBaris+5,KOLOM-11)	G(BARIS-
TengahBaris+5,KOLOM-10) G(BARIS-TengahBaris+5,KOLOM-9)...	
G(BARIS-TengahBaris+5,KOLOM-8)	G(BARIS-
TengahBaris+5,KOLOM-7) G(BARIS-TengahBaris+5,KOLOM-6)...	
G(BARIS-TengahBaris+5,KOLOM-4)...	
G(BARIS-TengahBaris+5,KOLOM-3)	G(BARIS-
TengahBaris+5,KOLOM-2) G(BARIS-TengahBaris+5,KOLOM-1) ...	
G(BARIS-TengahBaris+5,KOLOM-0)	G(BARIS-
TengahBaris+5,KOLOM+1) G(BARIS-TengahBaris+5,KOLOM+2) ...	
G(BARIS-TengahBaris+5,KOLOM+2)	G(BARIS-
TengahBaris+5,KOLOM+3) G(BARIS-TengahBaris+5,KOLOM+3) ...	
G(BARIS-TengahBaris+5,KOLOM+4)	G(BARIS-
TengahBaris+5,KOLOM+4) G(BARIS-TengahBaris+5,KOLOM+5) ...	
G(BARIS-TengahBaris+5,KOLOM+6)	G(BARIS-

TengahBaris+5,KOLOM+7)	G(BARIS-TengahBaris+5,KOLOM+8)...	
G(BARIS-TengahBaris+5,KOLOM+9)		G(BARIS-
TengahBaris+5,KOLOM+10)	G(BARIS-TengahBaris+5,KOLOM+11)...	
G(BARIS-TengahBaris+5,KOLOM+12)		G(BARIS-
TengahBaris+5,KOLOM+13)	G(BARIS-TengahBaris+5,KOLOM+14)...	
G(BARIS-TengahBaris+5,KOLOM+15)		G(BARIS-
TengahBaris+5,KOLOM+16)	G(BARIS-TengahBaris+5,KOLOM+17)...	
G(BARIS-TengahBaris+5,KOLOM+18)		G(BARIS-
TengahBaris+5,KOLOM+19)	G(BARIS-TengahBaris+5,KOLOM+20)...	
G(BARIS-TengahBaris+5,KOLOM+21)		G(BARIS-
TengahBaris+5,KOLOM+22);		
G(BARIS-TengahBaris+6,KOLOM-21)...		
G(BARIS-TengahBaris+6,KOLOM-20)		G(BARIS-
TengahBaris+6,KOLOM-19)	G(BARIS-TengahBaris+6,KOLOM-18)...	
G(BARIS-TengahBaris+6,KOLOM-17)		G(BARIS-
TengahBaris+6,KOLOM-16)	G(BARIS-TengahBaris+6,KOLOM-15)...	
G(BARIS-TengahBaris+6,KOLOM-14)		G(BARIS-
TengahBaris+6,KOLOM-13)	G(BARIS-TengahBaris+6,KOLOM-12)...	
G(BARIS-TengahBaris+6,KOLOM-11)		G(BARIS-
TengahBaris+6,KOLOM-10)	G(BARIS-TengahBaris+6,KOLOM-9)...	
G(BARIS-TengahBaris+6,KOLOM-8)		G(BARIS-
TengahBaris+6,KOLOM-7)	G(BARIS-TengahBaris+6,KOLOM-6)...	
G(BARIS-TengahBaris+6,KOLOM-4)...		
G(BARIS-TengahBaris+6,KOLOM-3)		G(BARIS-
TengahBaris+6,KOLOM-2)	G(BARIS-TengahBaris+6,KOLOM-1) ...	
G(BARIS-TengahBaris+6,KOLOM-0)		G(BARIS-
TengahBaris+6,KOLOM+1)	G(BARIS-TengahBaris+6,KOLOM+2) ...	
G(BARIS-TengahBaris+6,KOLOM+2)		G(BARIS-
TengahBaris+6,KOLOM+3)	G(BARIS-TengahBaris+6,KOLOM+3) ...	
G(BARIS-TengahBaris+6,KOLOM+4)		G(BARIS-
TengahBaris+6,KOLOM+4)	G(BARIS-TengahBaris+6,KOLOM+5) ...	
G(BARIS-TengahBaris+6,KOLOM+6)		G(BARIS-
TengahBaris+6,KOLOM+7)	G(BARIS-TengahBaris+6,KOLOM+8)...	
G(BARIS-TengahBaris+6,KOLOM+9)		G(BARIS-
TengahBaris+6,KOLOM+10)	G(BARIS-TengahBaris+6,KOLOM+11)...	
G(BARIS-TengahBaris+6,KOLOM+12)		G(BARIS-
TengahBaris+6,KOLOM+13)	G(BARIS-TengahBaris+6,KOLOM+14)...	
G(BARIS-TengahBaris+6,KOLOM+15)		G(BARIS-
TengahBaris+6,KOLOM+16)	G(BARIS-TengahBaris+6,KOLOM+17)...	
G(BARIS-TengahBaris+6,KOLOM+18)		G(BARIS-
TengahBaris+6,KOLOM+19)	G(BARIS-TengahBaris+6,KOLOM+20)...	
G(BARIS-TengahBaris+6,KOLOM+21)		G(BARIS-

TengahBaris+6,KOLOM+22);		
G(BARIS-TengahBaris+7,KOLOM-21)...		
G(BARIS-TengahBaris+7,KOLOM-20)		G(BARIS-
TengahBaris+7,KOLOM-19)	G(BARIS-TengahBaris+7,KOLOM-18)...	
G(BARIS-TengahBaris+7,KOLOM-17)		G(BARIS-
TengahBaris+7,KOLOM-16)	G(BARIS-TengahBaris+7,KOLOM-15)...	
G(BARIS-TengahBaris+7,KOLOM-14)		G(BARIS-
TengahBaris+7,KOLOM-13)	G(BARIS-TengahBaris+7,KOLOM-12)...	
G(BARIS-TengahBaris+7,KOLOM-11)		G(BARIS-
TengahBaris+7,KOLOM-10)	G(BARIS-TengahBaris+7,KOLOM-9)...	
G(BARIS-TengahBaris+7,KOLOM-8)		G(BARIS-
TengahBaris+7,KOLOM-7)	G(BARIS-TengahBaris+7,KOLOM-6)...	
G(BARIS-TengahBaris+7,KOLOM-4)...		
G(BARIS-TengahBaris+7,KOLOM-3)		G(BARIS-
TengahBaris+7,KOLOM-2)	G(BARIS-TengahBaris+7,KOLOM-1) ...	
G(BARIS-TengahBaris+7,KOLOM-0)		G(BARIS-
TengahBaris+7,KOLOM+1)	G(BARIS-TengahBaris+7,KOLOM+2) ...	
G(BARIS-TengahBaris+7,KOLOM+2)		G(BARIS-
TengahBaris+7,KOLOM+3)	G(BARIS-TengahBaris+7,KOLOM+3) ...	
G(BARIS-TengahBaris+7,KOLOM+4)		G(BARIS-
TengahBaris+7,KOLOM+4)	G(BARIS-TengahBaris+7,KOLOM+5) ...	
G(BARIS-TengahBaris+7,KOLOM+6)		G(BARIS-
TengahBaris+7,KOLOM+7)	G(BARIS-TengahBaris+7,KOLOM+8)...	
G(BARIS-TengahBaris+7,KOLOM+9)		G(BARIS-
TengahBaris+7,KOLOM+10)	G(BARIS-TengahBaris+7,KOLOM+11)...	
G(BARIS-TengahBaris+7,KOLOM+12)		G(BARIS-
TengahBaris+7,KOLOM+13)	G(BARIS-TengahBaris+7,KOLOM+14)...	
G(BARIS-TengahBaris+7,KOLOM+15)		G(BARIS-
TengahBaris+7,KOLOM+16)	G(BARIS-TengahBaris+7,KOLOM+17)...	
G(BARIS-TengahBaris+7,KOLOM+18)		G(BARIS-
TengahBaris+7,KOLOM+19)	G(BARIS-TengahBaris+7,KOLOM+20)...	
G(BARIS-TengahBaris+7,KOLOM+21)		G(BARIS-
TengahBaris+7,KOLOM+22);		
G(BARIS-TengahBaris+8,KOLOM-21)...		
G(BARIS-TengahBaris+8,KOLOM-20)		G(BARIS-
TengahBaris+8,KOLOM-19)	G(BARIS-TengahBaris+8,KOLOM-18)...	
G(BARIS-TengahBaris+8,KOLOM-17)		G(BARIS-
TengahBaris+8,KOLOM-16)	G(BARIS-TengahBaris+8,KOLOM-15)...	
G(BARIS-TengahBaris+8,KOLOM-14)		G(BARIS-
TengahBaris+8,KOLOM-13)	G(BARIS-TengahBaris+8,KOLOM-12)...	
G(BARIS-TengahBaris+8,KOLOM-11)		G(BARIS-



G(BARIS-TengahBaris+9,KOLOM+4)	G(BARIS-TengahBaris+9,KOLOM+4)	G(BARIS-TengahBaris+9,KOLOM+5) ...
G(BARIS-TengahBaris+9,KOLOM+6)	G(BARIS-TengahBaris+9,KOLOM+7)	G(BARIS-TengahBaris+9,KOLOM+8)...
G(BARIS-TengahBaris+9,KOLOM+9)	G(BARIS-TengahBaris+9,KOLOM+10)	G(BARIS-TengahBaris+9,KOLOM+11)...
G(BARIS-TengahBaris+9,KOLOM+12)	G(BARIS-TengahBaris+9,KOLOM+13)	G(BARIS-TengahBaris+9,KOLOM+14)...
G(BARIS-TengahBaris+9,KOLOM+15)	G(BARIS-TengahBaris+9,KOLOM+16)	G(BARIS-TengahBaris+9,KOLOM+17)...
G(BARIS-TengahBaris+9,KOLOM+18)	G(BARIS-TengahBaris+9,KOLOM+19)	G(BARIS-TengahBaris+9,KOLOM+20)...
G(BARIS-TengahBaris+9,KOLOM+21)	G(BARIS-TengahBaris+9,KOLOM+22);	
G(BARIS-TengahBaris+10,KOLOM-21)...		
G(BARIS-TengahBaris+10,KOLOM-20)	G(BARIS-TengahBaris+10,KOLOM-19)	G(BARIS-TengahBaris+10,KOLOM-18)...
G(BARIS-TengahBaris+10,KOLOM-17)	G(BARIS-TengahBaris+10,KOLOM-16)	G(BARIS-TengahBaris+10,KOLOM-15)...
G(BARIS-TengahBaris+10,KOLOM-14)	G(BARIS-TengahBaris+10,KOLOM-13)	G(BARIS-TengahBaris+10,KOLOM-12)...
G(BARIS-TengahBaris+10,KOLOM-11)	G(BARIS-TengahBaris+10,KOLOM-10)	G(BARIS-TengahBaris+10,KOLOM-9)...
G(BARIS-TengahBaris+10,KOLOM-8)	G(BARIS-TengahBaris+10,KOLOM-7)	G(BARIS-TengahBaris+10,KOLOM-6)...
G(BARIS-TengahBaris+10,KOLOM-4)...		
G(BARIS-TengahBaris+10,KOLOM-3)	G(BARIS-TengahBaris+10,KOLOM-2)	G(BARIS-TengahBaris+10,KOLOM-1) ...
G(BARIS-TengahBaris+10,KOLOM-0)	G(BARIS-TengahBaris+10,KOLOM+1)	G(BARIS-TengahBaris+10,KOLOM+2) ...
G(BARIS-TengahBaris+10,KOLOM+2)	G(BARIS-TengahBaris+10,KOLOM+3)	G(BARIS-TengahBaris+10,KOLOM+4) ...
G(BARIS-TengahBaris+10,KOLOM+4)	G(BARIS-TengahBaris+10,KOLOM+5)	G(BARIS-TengahBaris+10,KOLOM+6) ...
G(BARIS-TengahBaris+10,KOLOM+7)	G(BARIS-TengahBaris+10,KOLOM+8)	G(BARIS-TengahBaris+10,KOLOM+9) ...
G(BARIS-TengahBaris+10,KOLOM+9)	G(BARIS-TengahBaris+10,KOLOM+10)	G(BARIS-TengahBaris+10,KOLOM+11)...
G(BARIS-TengahBaris+10,KOLOM+12)	G(BARIS-TengahBaris+10,KOLOM+13)	G(BARIS-TengahBaris+10,KOLOM+14)...
G(BARIS-TengahBaris+10,KOLOM+15)	G(BARIS-TengahBaris+10,KOLOM+16)	G(BARIS-TengahBaris+10,KOLOM+17)...

G(BARIS-TengahBaris+10,KOLOM+18)	G(BARIS-TengahBaris+10,KOLOM+19)	G(BARIS-TengahBaris+10,KOLOM+20)...
G(BARIS-TengahBaris+10,KOLOM+21)	G(BARIS-TengahBaris+10,KOLOM+22);	
G(BARIS-TengahBaris+11,KOLOM-21)...	G(BARIS-TengahBaris+11,KOLOM-20)	G(BARIS-TengahBaris+11,KOLOM-19)
G(BARIS-TengahBaris+11,KOLOM-18)...	G(BARIS-TengahBaris+11,KOLOM-17)	G(BARIS-TengahBaris+11,KOLOM-16)
G(BARIS-TengahBaris+11,KOLOM-15)...	G(BARIS-TengahBaris+11,KOLOM-14)	G(BARIS-TengahBaris+11,KOLOM-13)
G(BARIS-TengahBaris+11,KOLOM-12)...	G(BARIS-TengahBaris+11,KOLOM-11)	G(BARIS-TengahBaris+11,KOLOM-10)
G(BARIS-TengahBaris+11,KOLOM-9)...	G(BARIS-TengahBaris+11,KOLOM-8)	G(BARIS-TengahBaris+11,KOLOM-7)
G(BARIS-TengahBaris+11,KOLOM-6)...	G(BARIS-TengahBaris+11,KOLOM-4)...	G(BARIS-TengahBaris+11,KOLOM-3)
G(BARIS-TengahBaris+11,KOLOM-2)	G(BARIS-TengahBaris+11,KOLOM-1) ...	G(BARIS-TengahBaris+11,KOLOM-0)
G(BARIS-TengahBaris+11,KOLOM+1)	G(BARIS-TengahBaris+11,KOLOM+2)	G(BARIS-TengahBaris+11,KOLOM+3)
G(BARIS-TengahBaris+11,KOLOM+4)	G(BARIS-TengahBaris+11,KOLOM+5) ...	G(BARIS-TengahBaris+11,KOLOM+6)
G(BARIS-TengahBaris+11,KOLOM+7)	G(BARIS-TengahBaris+11,KOLOM+8)...	G(BARIS-TengahBaris+11,KOLOM+9)
G(BARIS-TengahBaris+11,KOLOM+10)	G(BARIS-TengahBaris+11,KOLOM+11)...	G(BARIS-TengahBaris+11,KOLOM+12)
G(BARIS-TengahBaris+11,KOLOM+13)	G(BARIS-TengahBaris+11,KOLOM+14)...	G(BARIS-TengahBaris+11,KOLOM+15)
G(BARIS-TengahBaris+11,KOLOM+16)	G(BARIS-TengahBaris+11,KOLOM+17)...	G(BARIS-TengahBaris+11,KOLOM+18)
G(BARIS-TengahBaris+11,KOLOM+19)	G(BARIS-TengahBaris+11,KOLOM+20)...	G(BARIS-TengahBaris+11,KOLOM+21)
G(BARIS-TengahBaris+11,KOLOM+22);		
G(BARIS-TengahBaris+12,KOLOM-21)...	G(BARIS-TengahBaris+12,KOLOM-20)	G(BARIS-TengahBaris+12,KOLOM-19)
G(BARIS-TengahBaris+12,KOLOM-18)	G(BARIS-TengahBaris+12,KOLOM-17)	G(BARIS-TengahBaris+12,KOLOM-16)
G(BARIS-TengahBaris+12,KOLOM-15)...		





TengahBaris+13,KOLOM+1)	G(BARIS-TengahBaris+13,KOLOM+2) ...
G(BARIS-TengahBaris+13,KOLOM+2)	G(BARIS-TengahBaris+13,KOLOM+3) ...
TengahBaris+13,KOLOM+3)	G(BARIS-TengahBaris+13,KOLOM+4) ...
G(BARIS-TengahBaris+13,KOLOM+4)	G(BARIS-TengahBaris+13,KOLOM+5) ...
TengahBaris+13,KOLOM+4)	G(BARIS-TengahBaris+13,KOLOM+6) ...
G(BARIS-TengahBaris+13,KOLOM+6)	G(BARIS-TengahBaris+13,KOLOM+7) ...
TengahBaris+13,KOLOM+7)	G(BARIS-TengahBaris+13,KOLOM+8)...
G(BARIS-TengahBaris+13,KOLOM+9)	G(BARIS-TengahBaris+13,KOLOM+10) ...
TengahBaris+13,KOLOM+10)	G(BARIS-TengahBaris+13,KOLOM+11)...
G(BARIS-TengahBaris+13,KOLOM+12)	G(BARIS-TengahBaris+13,KOLOM+13) ...
TengahBaris+13,KOLOM+13)	G(BARIS-TengahBaris+13,KOLOM+14)...
G(BARIS-TengahBaris+13,KOLOM+15)	G(BARIS-TengahBaris+13,KOLOM+16) ...
TengahBaris+13,KOLOM+16)	G(BARIS-TengahBaris+13,KOLOM+17)...
G(BARIS-TengahBaris+13,KOLOM+18)	G(BARIS-TengahBaris+13,KOLOM+19) ...
TengahBaris+13,KOLOM+19)	G(BARIS-TengahBaris+13,KOLOM+20)...
G(BARIS-TengahBaris+13,KOLOM+21)	G(BARIS-TengahBaris+13,KOLOM+22);
G(BARIS-TengahBaris+14,KOLOM-21)...	G(BARIS-TengahBaris+14,KOLOM-20) ...
TengahBaris+14,KOLOM-19)	G(BARIS-TengahBaris+14,KOLOM-18)...
G(BARIS-TengahBaris+14,KOLOM-17)	G(BARIS-TengahBaris+14,KOLOM-16) ...
TengahBaris+14,KOLOM-16)	G(BARIS-TengahBaris+14,KOLOM-15)...
G(BARIS-TengahBaris+14,KOLOM-14)	G(BARIS-TengahBaris+14,KOLOM-13) ...
TengahBaris+14,KOLOM-13)	G(BARIS-TengahBaris+14,KOLOM-12)...
G(BARIS-TengahBaris+14,KOLOM-11)	G(BARIS-TengahBaris+14,KOLOM-10) ...
TengahBaris+14,KOLOM-10)	G(BARIS-TengahBaris+14,KOLOM-9)...
G(BARIS-TengahBaris+14,KOLOM-8)	G(BARIS-TengahBaris+14,KOLOM-7) ...
TengahBaris+14,KOLOM-7)	G(BARIS-TengahBaris+14,KOLOM-6)...
G(BARIS-TengahBaris+14,KOLOM-4)...	G(BARIS-TengahBaris+14,KOLOM-3) ...
TengahBaris+14,KOLOM-2)	G(BARIS-TengahBaris+14,KOLOM-1) ...
G(BARIS-TengahBaris+14,KOLOM-0)	G(BARIS-TengahBaris+14,KOLOM+1) ...
TengahBaris+14,KOLOM+1)	G(BARIS-TengahBaris+14,KOLOM+2) ...
G(BARIS-TengahBaris+14,KOLOM+2)	G(BARIS-TengahBaris+14,KOLOM+3) ...
TengahBaris+14,KOLOM+3)	G(BARIS-TengahBaris+14,KOLOM+4) ...
G(BARIS-TengahBaris+14,KOLOM+4)	G(BARIS-TengahBaris+14,KOLOM+5) ...
TengahBaris+14,KOLOM+4)	G(BARIS-TengahBaris+14,KOLOM+6) ...
G(BARIS-TengahBaris+14,KOLOM+6)	G(BARIS-TengahBaris+14,KOLOM+7) ...
TengahBaris+14,KOLOM+7)	G(BARIS-TengahBaris+14,KOLOM+8)...
G(BARIS-TengahBaris+14,KOLOM+9)	G(BARIS-TengahBaris+14,KOLOM+10) ...
TengahBaris+14,KOLOM+10)	G(BARIS-TengahBaris+14,KOLOM+11)...
G(BARIS-TengahBaris+14,KOLOM+12)	G(BARIS-TengahBaris+14,KOLOM+13) ...

TengahBaris+14,KOLOM+13)	G(BARIS-TengahBaris+14,KOLOM+14)...	
G(BARIS-TengahBaris+14,KOLOM+15)		G(BARIS-
TengahBaris+14,KOLOM+16)	G(BARIS-TengahBaris+14,KOLOM+17)...	
G(BARIS-TengahBaris+14,KOLOM+18)		G(BARIS-
TengahBaris+14,KOLOM+19)	G(BARIS-TengahBaris+14,KOLOM+20)...	
G(BARIS-TengahBaris+14,KOLOM+21)		G(BARIS-
TengahBaris+14,KOLOM+22);		
G(BARIS-TengahBaris+15,KOLOM-21)...		
G(BARIS-TengahBaris+15,KOLOM-20)		G(BARIS-
TengahBaris+15,KOLOM-19)	G(BARIS-TengahBaris+15,KOLOM-18)...	
G(BARIS-TengahBaris+15,KOLOM-17)		G(BARIS-
TengahBaris+15,KOLOM-16)	G(BARIS-TengahBaris+15,KOLOM-15)...	
G(BARIS-TengahBaris+15,KOLOM-14)		G(BARIS-
TengahBaris+15,KOLOM-13)	G(BARIS-TengahBaris+15,KOLOM-12)...	
G(BARIS-TengahBaris+15,KOLOM-11)		G(BARIS-
TengahBaris+15,KOLOM-10)	G(BARIS-TengahBaris+15,KOLOM-9)...	
G(BARIS-TengahBaris+15,KOLOM-8)		G(BARIS-
TengahBaris+15,KOLOM-7)	G(BARIS-TengahBaris+15,KOLOM-6)...	
G(BARIS-TengahBaris+15,KOLOM-4)...		
G(BARIS-TengahBaris+15,KOLOM-3)		G(BARIS-
TengahBaris+15,KOLOM-2)	G(BARIS-TengahBaris+15,KOLOM-1) ...	
G(BARIS-TengahBaris+15,KOLOM-0)		G(BARIS-
TengahBaris+15,KOLOM+1)	G(BARIS-TengahBaris+15,KOLOM+2) ...	
G(BARIS-TengahBaris+15,KOLOM+2)		G(BARIS-
TengahBaris+15,KOLOM+3)	G(BARIS-TengahBaris+15,KOLOM+3) ...	
G(BARIS-TengahBaris+15,KOLOM+4)		G(BARIS-
TengahBaris+15,KOLOM+4)	G(BARIS-TengahBaris+15,KOLOM+5) ...	
G(BARIS-TengahBaris+15,KOLOM+6)		G(BARIS-
TengahBaris+15,KOLOM+7)	G(BARIS-TengahBaris+15,KOLOM+8)...	
G(BARIS-TengahBaris+15,KOLOM+9)		G(BARIS-
TengahBaris+15,KOLOM+10)	G(BARIS-TengahBaris+15,KOLOM+11)...	
G(BARIS-TengahBaris+15,KOLOM+12)		G(BARIS-
TengahBaris+15,KOLOM+13)	G(BARIS-TengahBaris+15,KOLOM+14)...	
G(BARIS-TengahBaris+15,KOLOM+15)		G(BARIS-
TengahBaris+15,KOLOM+16)	G(BARIS-TengahBaris+15,KOLOM+17)...	
G(BARIS-TengahBaris+15,KOLOM+18)		G(BARIS-
TengahBaris+15,KOLOM+19)	G(BARIS-TengahBaris+15,KOLOM+20)...	
G(BARIS-TengahBaris+15,KOLOM+21)		G(BARIS-
TengahBaris+15,KOLOM+22);		
G(BARIS-TengahBaris+16,KOLOM-21)...		
G(BARIS-TengahBaris+16,KOLOM-20)		G(BARIS-

TengahBaris+16,KOLOM-19)	G(BARIS-TengahBaris+16,KOLOM-18)...
G(BARIS-TengahBaris+16,KOLOM-17)	G(BARIS-TengahBaris+16,KOLOM-16)
G(BARIS-TengahBaris+16,KOLOM-14)	G(BARIS-TengahBaris+16,KOLOM-13)
G(BARIS-TengahBaris+16,KOLOM-11)	G(BARIS-TengahBaris+16,KOLOM-10)
G(BARIS-TengahBaris+16,KOLOM-8)	G(BARIS-TengahBaris+16,KOLOM-7)
G(BARIS-TengahBaris+16,KOLOM-4)...	G(BARIS-TengahBaris+16,KOLOM-3)
G(BARIS-TengahBaris+16,KOLOM-2)	G(BARIS-TengahBaris+16,KOLOM-1) ...
G(BARIS-TengahBaris+16,KOLOM-0)	G(BARIS-TengahBaris+16,KOLOM+1)
G(BARIS-TengahBaris+16,KOLOM+2)	G(BARIS-TengahBaris+16,KOLOM+3)
G(BARIS-TengahBaris+16,KOLOM+4)	G(BARIS-TengahBaris+16,KOLOM+5) ...
G(BARIS-TengahBaris+16,KOLOM+6)	G(BARIS-TengahBaris+16,KOLOM+7)
G(BARIS-TengahBaris+16,KOLOM+9)	G(BARIS-TengahBaris+16,KOLOM+10)
G(BARIS-TengahBaris+16,KOLOM+12)	G(BARIS-TengahBaris+16,KOLOM+13)
G(BARIS-TengahBaris+16,KOLOM+15)	G(BARIS-TengahBaris+16,KOLOM+16)
G(BARIS-TengahBaris+16,KOLOM+18)	G(BARIS-TengahBaris+16,KOLOM+19)
G(BARIS-TengahBaris+16,KOLOM+21)	G(BARIS-TengahBaris+16,KOLOM+22);
G(BARIS-TengahBaris+17,KOLOM-21)...	G(BARIS-TengahBaris+17,KOLOM-20)
G(BARIS-TengahBaris+17,KOLOM-17)	G(BARIS-TengahBaris+17,KOLOM-16)
G(BARIS-TengahBaris+17,KOLOM-14)	G(BARIS-TengahBaris+17,KOLOM-13)
G(BARIS-TengahBaris+17,KOLOM-11)	G(BARIS-TengahBaris+17,KOLOM-10)
G(BARIS-TengahBaris+17,KOLOM-8)	G(BARIS-TengahBaris+17,KOLOM-7)
G(BARIS-TengahBaris+17,KOLOM-4)...	

G(BARIS-TengahBaris+17,KOLOM-3)	G(BARIS-TengahBaris+17,KOLOM-2)	G(BARIS-TengahBaris+17,KOLOM-1) ...
G(BARIS-TengahBaris+17,KOLOM-0)	G(BARIS-TengahBaris+17,KOLOM+1)	G(BARIS-TengahBaris+17,KOLOM+2) ...
G(BARIS-TengahBaris+17,KOLOM+2)	G(BARIS-TengahBaris+17,KOLOM+3)	G(BARIS-TengahBaris+17,KOLOM+4) ...
G(BARIS-TengahBaris+17,KOLOM+4)	G(BARIS-TengahBaris+17,KOLOM+5)	G(BARIS-TengahBaris+17,KOLOM+6) ...
G(BARIS-TengahBaris+17,KOLOM+6)	G(BARIS-TengahBaris+17,KOLOM+7)	G(BARIS-TengahBaris+17,KOLOM+8)...
G(BARIS-TengahBaris+17,KOLOM+9)	G(BARIS-TengahBaris+17,KOLOM+10)	G(BARIS-TengahBaris+17,KOLOM+11)...
G(BARIS-TengahBaris+17,KOLOM+12)	G(BARIS-TengahBaris+17,KOLOM+13)	G(BARIS-TengahBaris+17,KOLOM+14)...
G(BARIS-TengahBaris+17,KOLOM+15)	G(BARIS-TengahBaris+17,KOLOM+16)	G(BARIS-TengahBaris+17,KOLOM+17)...
G(BARIS-TengahBaris+17,KOLOM+18)	G(BARIS-TengahBaris+17,KOLOM+19)	G(BARIS-TengahBaris+17,KOLOM+20)...
G(BARIS-TengahBaris+17,KOLOM+21)	G(BARIS-TengahBaris+17,KOLOM+22);	
G(BARIS-TengahBaris+18,KOLOM-21)...	G(BARIS-TengahBaris+18,KOLOM-20)	G(BARIS-TengahBaris+18,KOLOM-19)
G(BARIS-TengahBaris+18,KOLOM-19)	G(BARIS-TengahBaris+18,KOLOM-18)...	G(BARIS-TengahBaris+18,KOLOM-17)
G(BARIS-TengahBaris+18,KOLOM-17)	G(BARIS-TengahBaris+18,KOLOM-16)	G(BARIS-TengahBaris+18,KOLOM-15)...
G(BARIS-TengahBaris+18,KOLOM-14)	G(BARIS-TengahBaris+18,KOLOM-13)	G(BARIS-TengahBaris+18,KOLOM-12)...
G(BARIS-TengahBaris+18,KOLOM-11)	G(BARIS-TengahBaris+18,KOLOM-10)	G(BARIS-TengahBaris+18,KOLOM-9)...
G(BARIS-TengahBaris+18,KOLOM-8)	G(BARIS-TengahBaris+18,KOLOM-7)	G(BARIS-TengahBaris+18,KOLOM-6)...
G(BARIS-TengahBaris+18,KOLOM-4)...	G(BARIS-TengahBaris+18,KOLOM-3)	G(BARIS-TengahBaris+18,KOLOM-2)
G(BARIS-TengahBaris+18,KOLOM-2)	G(BARIS-TengahBaris+18,KOLOM-1)	G(BARIS-TengahBaris+18,KOLOM-0)
G(BARIS-TengahBaris+18,KOLOM+1)	G(BARIS-TengahBaris+18,KOLOM+2)	G(BARIS-TengahBaris+18,KOLOM+3)
G(BARIS-TengahBaris+18,KOLOM+2)	G(BARIS-TengahBaris+18,KOLOM+3)	G(BARIS-TengahBaris+18,KOLOM+4)
G(BARIS-TengahBaris+18,KOLOM+4)	G(BARIS-TengahBaris+18,KOLOM+5)	G(BARIS-TengahBaris+18,KOLOM+6)
G(BARIS-TengahBaris+18,KOLOM+6)	G(BARIS-TengahBaris+18,KOLOM+7)	G(BARIS-TengahBaris+18,KOLOM+8)...

G(BARIS-TengahBaris+18,KOLOM+9)	G(BARIS-TengahBaris+18,KOLOM+10)	G(BARIS-TengahBaris+18,KOLOM+11)...
G(BARIS-TengahBaris+18,KOLOM+12)	G(BARIS-TengahBaris+18,KOLOM+13)	G(BARIS-TengahBaris+18,KOLOM+14)...
G(BARIS-TengahBaris+18,KOLOM+15)	G(BARIS-TengahBaris+18,KOLOM+16)	G(BARIS-TengahBaris+18,KOLOM+17)...
G(BARIS-TengahBaris+18,KOLOM+18)	G(BARIS-TengahBaris+18,KOLOM+19)	G(BARIS-TengahBaris+18,KOLOM+20)...
G(BARIS-TengahBaris+18,KOLOM+21)	G(BARIS-TengahBaris+18,KOLOM+22);	
G(BARIS-TengahBaris+19,KOLOM-21)...	G(BARIS-TengahBaris+19,KOLOM-20)	G(BARIS-TengahBaris+19,KOLOM-19)
G(BARIS-TengahBaris+19,KOLOM-18)	G(BARIS-TengahBaris+19,KOLOM-17)	G(BARIS-TengahBaris+19,KOLOM-16)
G(BARIS-TengahBaris+19,KOLOM-15)	G(BARIS-TengahBaris+19,KOLOM-14)	G(BARIS-TengahBaris+19,KOLOM-13)
G(BARIS-TengahBaris+19,KOLOM-12)	G(BARIS-TengahBaris+19,KOLOM-11)	G(BARIS-TengahBaris+19,KOLOM-10)
G(BARIS-TengahBaris+19,KOLOM-9)	G(BARIS-TengahBaris+19,KOLOM-8)	G(BARIS-TengahBaris+19,KOLOM-7)
G(BARIS-TengahBaris+19,KOLOM-6)	G(BARIS-TengahBaris+19,KOLOM-4)...	G(BARIS-TengahBaris+19,KOLOM-3)
G(BARIS-TengahBaris+19,KOLOM-2)	G(BARIS-TengahBaris+19,KOLOM-1) ...	G(BARIS-TengahBaris+19,KOLOM-0)
G(BARIS-TengahBaris+19,KOLOM+1)	G(BARIS-TengahBaris+19,KOLOM+2) ...	G(BARIS-TengahBaris+19,KOLOM+3)
G(BARIS-TengahBaris+19,KOLOM+4)	G(BARIS-TengahBaris+19,KOLOM+5) ...	G(BARIS-TengahBaris+19,KOLOM+6)
G(BARIS-TengahBaris+19,KOLOM+7)	G(BARIS-TengahBaris+19,KOLOM+8)...	G(BARIS-TengahBaris+19,KOLOM+9)
G(BARIS-TengahBaris+19,KOLOM+10)	G(BARIS-TengahBaris+19,KOLOM+11)...	G(BARIS-TengahBaris+19,KOLOM+12)
G(BARIS-TengahBaris+19,KOLOM+13)	G(BARIS-TengahBaris+19,KOLOM+14)...	G(BARIS-TengahBaris+19,KOLOM+15)
G(BARIS-TengahBaris+19,KOLOM+16)	G(BARIS-TengahBaris+19,KOLOM+17)...	G(BARIS-TengahBaris+19,KOLOM+18)
G(BARIS-TengahBaris+19,KOLOM+19)	G(BARIS-TengahBaris+19,KOLOM+20)...	G(BARIS-TengahBaris+19,KOLOM+21)
G(BARIS-TengahBaris+19,KOLOM+22);		

G(BARIS-TengahBaris+20,KOLOM-21)...	
G(BARIS-TengahBaris+20,KOLOM-20)	G(BARIS-TengahBaris+20,KOLOM-19) G(BARIS-TengahBaris+20,KOLOM-18)...
G(BARIS-TengahBaris+20,KOLOM-17)	G(BARIS-TengahBaris+20,KOLOM-16) G(BARIS-TengahBaris+20,KOLOM-15)...
G(BARIS-TengahBaris+20,KOLOM-14)	G(BARIS-TengahBaris+20,KOLOM-13) G(BARIS-TengahBaris+20,KOLOM-12)...
G(BARIS-TengahBaris+20,KOLOM-11)	G(BARIS-TengahBaris+20,KOLOM-10) G(BARIS-TengahBaris+20,KOLOM-9)...
G(BARIS-TengahBaris+20,KOLOM-8)	G(BARIS-TengahBaris+20,KOLOM-7) G(BARIS-TengahBaris+20,KOLOM-6)...
G(BARIS-TengahBaris+20,KOLOM-4)...	
G(BARIS-TengahBaris+20,KOLOM-3)	G(BARIS-TengahBaris+20,KOLOM-2) G(BARIS-TengahBaris+20,KOLOM-1) ...
G(BARIS-TengahBaris+20,KOLOM-0)	G(BARIS-TengahBaris+20,KOLOM+1) G(BARIS-TengahBaris+20,KOLOM+2) ...
G(BARIS-TengahBaris+20,KOLOM+2)	G(BARIS-TengahBaris+20,KOLOM+3) G(BARIS-TengahBaris+20,KOLOM+3) ...
G(BARIS-TengahBaris+20,KOLOM+4)	G(BARIS-TengahBaris+20,KOLOM+4) G(BARIS-TengahBaris+20,KOLOM+5) ...
G(BARIS-TengahBaris+20,KOLOM+6)	G(BARIS-TengahBaris+20,KOLOM+7) G(BARIS-TengahBaris+20,KOLOM+8)...
G(BARIS-TengahBaris+20,KOLOM+9)	G(BARIS-TengahBaris+20,KOLOM+10) G(BARIS-TengahBaris+20,KOLOM+11)...
G(BARIS-TengahBaris+20,KOLOM+12)	G(BARIS-TengahBaris+20,KOLOM+13) G(BARIS-TengahBaris+20,KOLOM+14)...
G(BARIS-TengahBaris+20,KOLOM+15)	G(BARIS-TengahBaris+20,KOLOM+16) G(BARIS-TengahBaris+20,KOLOM+17)...
G(BARIS-TengahBaris+20,KOLOM+18)	G(BARIS-TengahBaris+20,KOLOM+19) G(BARIS-TengahBaris+20,KOLOM+20)...
G(BARIS-TengahBaris+20,KOLOM+21)	G(BARIS-TengahBaris+20,KOLOM+22);
G(BARIS-TengahBaris+21,KOLOM-21)...	
G(BARIS-TengahBaris+21,KOLOM-20)	G(BARIS-TengahBaris+21,KOLOM-19) G(BARIS-TengahBaris+21,KOLOM-18)...
G(BARIS-TengahBaris+21,KOLOM-17)	G(BARIS-TengahBaris+21,KOLOM-16) G(BARIS-TengahBaris+21,KOLOM-15)...
G(BARIS-TengahBaris+21,KOLOM-14)	G(BARIS-TengahBaris+21,KOLOM-13) G(BARIS-TengahBaris+21,KOLOM-12)...
G(BARIS-TengahBaris+21,KOLOM-11)	G(BARIS-TengahBaris+21,KOLOM-10) G(BARIS-TengahBaris+21,KOLOM-9)...

```

        G(BARIS-TengahBaris+21,KOLOM-8)                G(BARIS-
TengahBaris+21,KOLOM-7) G(BARIS-TengahBaris+21,KOLOM-6)...
        G(BARIS-TengahBaris+21,KOLOM-4)...
        G(BARIS-TengahBaris+21,KOLOM-3)                G(BARIS-
TengahBaris+21,KOLOM-2) G(BARIS-TengahBaris+21,KOLOM-1) ...
        G(BARIS-TengahBaris+21,KOLOM-0)                G(BARIS-
TengahBaris+21,KOLOM+1) G(BARIS-TengahBaris+21,KOLOM+2) ...
        G(BARIS-TengahBaris+21,KOLOM+2)                G(BARIS-
TengahBaris+21,KOLOM+3) G(BARIS-TengahBaris+21,KOLOM+3) ...
        G(BARIS-TengahBaris+21,KOLOM+4)                G(BARIS-
TengahBaris+21,KOLOM+4) G(BARIS-TengahBaris+21,KOLOM+5) ...
        G(BARIS-TengahBaris+21,KOLOM+6)                G(BARIS-
TengahBaris+21,KOLOM+7) G(BARIS-TengahBaris+21,KOLOM+8)...
        G(BARIS-TengahBaris+21,KOLOM+9)                G(BARIS-
TengahBaris+21,KOLOM+10) G(BARIS-TengahBaris+21,KOLOM+11)...
        G(BARIS-TengahBaris+21,KOLOM+12)                G(BARIS-
TengahBaris+21,KOLOM+13) G(BARIS-TengahBaris+21,KOLOM+14)...
        G(BARIS-TengahBaris+21,KOLOM+15)                G(BARIS-
TengahBaris+21,KOLOM+16) G(BARIS-TengahBaris+21,KOLOM+17)...
        G(BARIS-TengahBaris+21,KOLOM+18)                G(BARIS-
TengahBaris+21,KOLOM+19) G(BARIS-TengahBaris+21,KOLOM+20)...
        G(BARIS-TengahBaris+21,KOLOM+21)                G(BARIS-
TengahBaris+21,KOLOM+22);];
    %sum(Asz)
    %sum(Asz)
    [tinggi,lebar]=size(Asz)
    ambang=10;
    for baris=1:tinggi
        for kolom=1:lebar
            if Asz(baris,kolom)>= ambang;
                Asz(baris,kolom)=1;
            else
                Asz(baris,kolom)=0;
            end
        end
    end
    end
    Asz;
Return
end

```



**Lampiran A.2** Program *Tresholding* ditulis ke dalam Matlab Function sebagai berikut.

```
function [r, g, b, bin] = clr_thr(red, ...
    green, blue, red_th, green_th, blue_th)
% This block supports the Embedded MATLAB subset.
% See the help menu for details.
red_vec = thresh(red, red_th, 1);
green_vec = thresh(green, green_th, 1);
blue_vec = thresh(blue, blue_th, 1);

tmp = double(and(red_vec, green_vec));
tmp_blob = uint8(and(tmp, blue_vec));
bin = uint8(255*tmp_blob);

% background = uint8(255*double(not(tmp_blob)));
% r = uint8(red.*tmp_blob + background);
% g = uint8(green.*tmp_blob + background);
% b = uint8(blue.*tmp_blob + background);
r = uint8(red.*tmp_blob);
g = uint8(green.*tmp_blob);
b = uint8(blue.*tmp_blob);

return;

%-----
function x = thresh(x, th, val)
[row col] = size(x);
for i=1:row
    for j=1:col
        if x(i,j) >= th(1) && x(i, j) <= th(2)
            x(i, j) = val;
        else
            x(i, j) = 0;
        end
    end
end
return;
%-----
```

**Lampiran A.3** Pengubahan matrik 3 Dimensi ke satu dimensi dengan mempertimbangkan warna biru sebagai warna yang dijadikan warna jalur.

```
function y = dilate(x, n)
    [row col] = size(x);
    se = ones(3, 3);
    tmp = double(x);
    if n>=1
        for i=1:n
            yp = conv2(tmp, se);
            yp = min(yp, 255);
            tmp = yp(2:row+1, 2:col+1);
        end
    end
    y = uint8(tmp);

    return;
```

List matlab program untuk menemukan titik target dalam bentuk pixel.

```
function [r, g, b, n] = find_blobs(BIN, type)
% This block supports the Embedded MATLAB subset.
% See the help menu for details.
idvec = zeros(1, 50);
[ROW COL] = size(BIN);
if ROW>120 || COL>160
    r = uint8(BIN);
    g = uint8(BIN);
    b = uint8(BIN);
    n = -1;
    return;
end
dwnsample = 4;
row = round(ROW/dwnsample);
col = round(COL/dwnsample);
bin = uint8(zeros(row, col));

rw = 0;
% for i=dwnsample+1:dwnsample:ROW
for i=1:dwnsample:ROW
```

```

    rw = rw + 1;
    co = 0;
    for j=1:dwnsmple:COL
        co = co + 1;
        bin(rw, co) = uint8(BIN(i, j));
    end
end

% r = uint8(zeros(row, col));
% g = uint8(zeros(row, col));
% b = uint8(zeros(row, col));
r = uint8(bin);
g = uint8(bin);
b = uint8(bin);
n = -1;%count;

% Bin = double(bin)/255;
% binsum = sum(Bin(:));
% ratio = binsum/(row*col);

% if ratio<=0.1
% Find the 4 or 8 connected reachability matrix
[eqtable, limg, maxlabel] = find_reachability_matrix(bin, type);
if maxlabel>=1
    % Find equivalent labels
    table1 = updatetable(eqtable, maxlabel);
    table2 = updatetable2(table1, maxlabel);
    table3 = double(or(table1, table2));
    [idvec, n] = updatetable3(table3, maxlabel);
%
%
%    % Label the final image
    Limg = label_image(limg, n, idvec, maxlabel);
    r = uint8(Limg);
    g = uint8(Limg);
    b = uint8(Limg);
end
%    r = uint8(limg);

```

```

%   g = uint8(limg);
%   b = uint8(limg);
% end

return;
%-----
function [eqtable, limg, maxlabel] = find_reachability_matrix(img,
Type)

[row col] = size(img);

type = Type;
if Type~=4 && Type~=8
    type = 8;
end

% Step 1: Label 8 or 4 connected components
limg = zeros(row, col);
label = 0;
Maxlabel = 50;
for i=1:row
    for j=1:col
        if img(i, j)>0 && limg(i, j)==0
            if (j-1)>=1 && img(i, j-1)>0
                limg(i, j)= limg(i, j-1);
            else
                if label<=(Maxlabel-1)
                    label = label+1;
                    limg(i, j)= label;
                else
                    limg(i, j)= 0;
                end
            end
        end
    end
end
end
end

% Find the reachability matrix

```

```

maxlabel = label;
eqtable = eye(Maxlabel, Maxlabel);
if maxlabel<1
    return;
end
% eqtable = eye(maxlabel, maxlabel);
for i=2:row
    for j=1:col
        if type == 4
            if img(i, j)>0
                if img(i-1, j)>0
                    eqtable(limg(i, j), limg(i-1, j)) = 1;
                    eqtable(limg(i-1, j), limg(i, j)) = 1;
                end
            end
        elseif type == 8
            if img(i, j)>0
                if (j-1)>=1 && img(i-1, j-1)>0
                    eqtable(limg(i, j), limg(i-1, j-1)) = 1;
                    eqtable(limg(i-1, j-1), limg(i, j)) = 1;
                end
                if img(i-1, j)>0
                    eqtable(limg(i, j), limg(i-1, j)) = 1;
                    eqtable(limg(i-1, j), limg(i, j)) = 1;
                end
                if (j+1)<=col && img(i-1, j+1)>0
                    eqtable(limg(i, j), limg(i-1, j+1)) = 1;
                    eqtable(limg(i-1, j+1), limg(i, j)) = 1;
                end
            end
        end
    end
end

end
end

return;
%-----
function [table] = updatetable(eqtable, maxlabel)

```

```

[row col] = size(ehtable);
table = zeros(row, col);
% ehtable = ehtable;
for i=1:maxlabel
    tmp1 = ehtable(i, :);
    for j=i+1:maxlabel
        tmp2 = ehtable(j, :);
        iscommon = sum(double(and(tmp1, tmp2)));
        if iscommon
            tmp1 = double(or(tmp1, tmp2));
        end
    end
    table(i, :) = tmp1;
end

return;
% -----
function [table] = updatetable2(ehtable, maxlabel)

[row col] = size(ehtable);
table = zeros(row, col);
% ehtable = ehtable;
for i=maxlabel:-1:1
    tmp1 = ehtable(i, :);
    %     i
    %     count = count + 1;
    %     idvec(i) = count;
    for j=i-1:-1:1
        tmp2 = ehtable(j, :);
        iscommon = sum(double(and(tmp1, tmp2)));
        if iscommon
            tmp1 = double(or(tmp1, tmp2));
        %         find(tmp1==1)
        %         pause
    end
end
table(i, :) = tmp1;

```

```

end

return;
%-----
function [idvec, count] = updatetable3(ehtable, maxlabel)

[row col] = size(ehtable);
table = zeros(row, col);
count = 0;
idvec = zeros(1, col);
% ehtable = ehtable;
for i=1:maxlabel
    tmp1 = ehtable(i, :);
    if tmp1(i)~=0
        count = count + 1;
        idvec(i) = count;
        for j=i+1:maxlabel
            tmp2 = ehtable(j, :);
            iscommon = sum(double(and(tmp1, tmp2)));
            if iscommon
                tmp1 = double(or(tmp1, tmp2));
            end
        end
        table(i, :) = tmp1;
        for j=i+1:maxlabel
            if tmp1(j)==1
                idvec(j) = count;
                ehtable(j, :) = table(j, :);
            end
        end
    end
end
end

return;
%-----
function [Limg] = label_image(limg, count, idvec, maxlabel)

[row col] = size(limg);

```

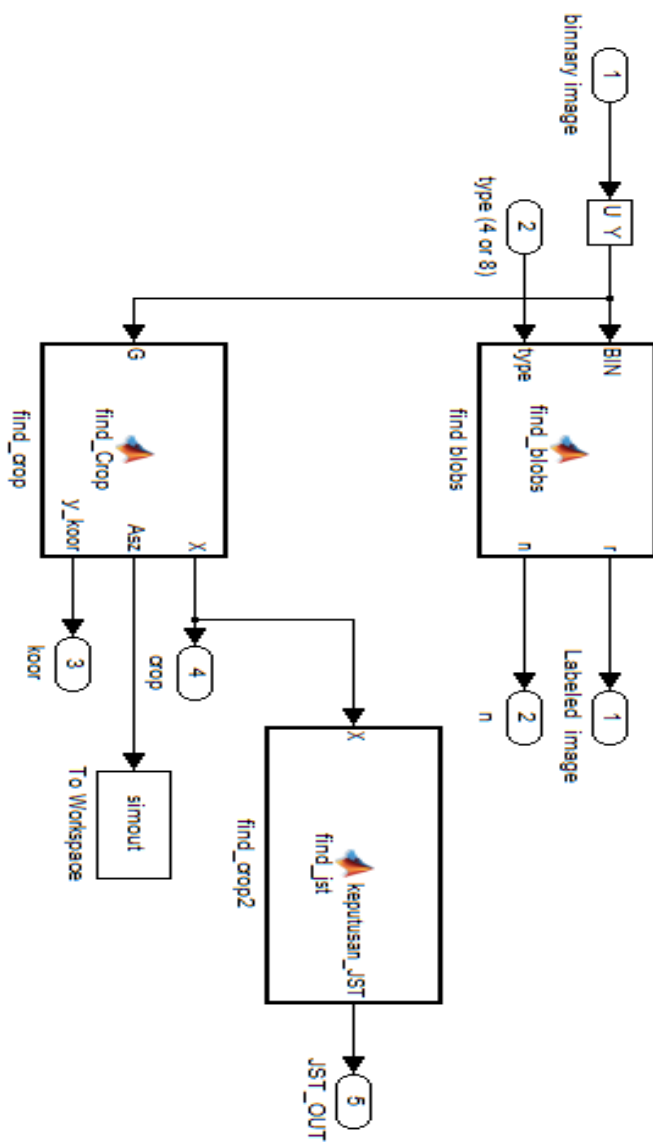
```

Limg = limg;
for i=1:row
    for j=1:col
        if Limg(i, j)>0 && Limg(i, j)<=maxlabel
            %         if count<=255
            %             Limg(i, j) = uint8(round((255/count)*idvec(Limg(i, j))));
            %         else
            %             Limg(i, j) = uint8(idvec(Limg(i, j)));
            %         end
        end
    end
end
return;
%-----

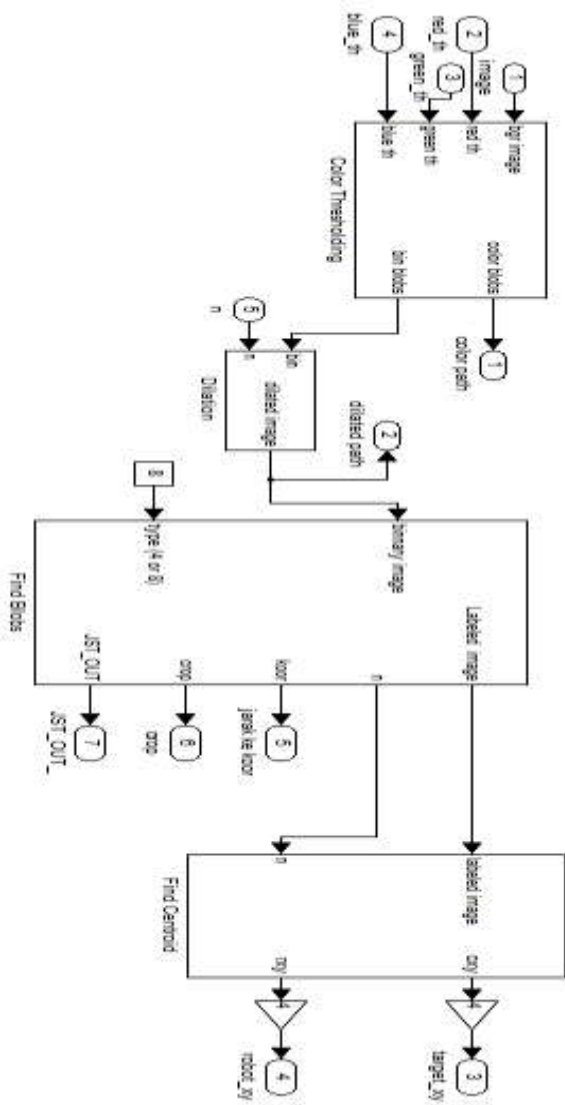
```



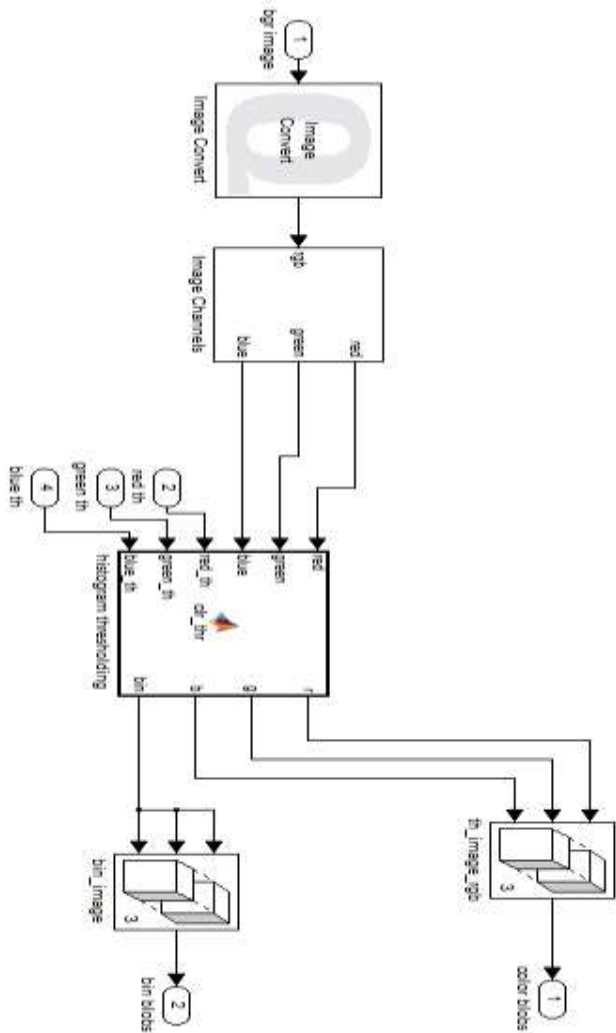
Lampiran B. 1 Gambar *Simulink Find\_Crop, Find\_Jst dan Find\_blobs*



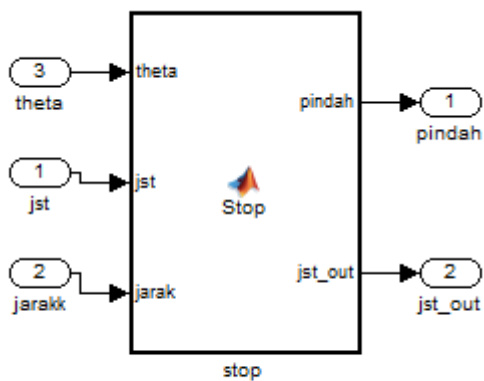
Lampiran B.2 Gambar *Simulink* Color\_Tesholding dan Find\_Blobs



Lampiran B.3 Gambar *Simulink histogram Image*



Lampiran B.4 Gambar *Simulink Switching Motor (Stop or not)*



,

*Halaman ini sengaja dikosongkan*

## BIODATA PENULIS



Fauzi Saputra, dilahirkan di Balai Selasa pada tanggal 04 Nopember 1992. Penulis merupakan anak keempat dari empat bersaudara. Penulis telah menempuh pendidikan formal di SDN 02 Ranah Pesisir, Pesisir Selatan, Sumatra Barat(1998-2004), SMPN 1 Ranah Pesisir, Pesisir Selatan, Sumatra Barat (2004-2007), SMAN 1 Ranah Pesisir, Pesisir Selatan, Sumatra Barat (2007-2010) dan D3 Jurusan Teknik Elektro dengan bidang studi Teknik Elektronika di Universitas Gadjah Mada (2011-2015). Selanjutnya melanjutkan ke jenjang strata satu di Fakultas Teknologi Elektro, Institut Teknologi Sepuluh Nopember Surabaya (2015-2017). Penulis mengambil konsentrasi bidang studi Teknik Sistem Pengaturan. Pada kesempatan menyelesaikan tugas akhir penulis mengambil judul **“Perancangan Jaringan Saraf Tiruan untuk Pengenalan Pola Pada Navigasi Mobile Robot Berbasis Data Gambar”**.

*Contact Person :*

*E-mail* : fauzi.yuskha@gmail.com

*Halaman ini sengaja dikosongkan*